Knowledge at First Glance: A Model for a Data Visualization Recommender System Suited for Non-expert Users

Petra Kubernátová[1], Magda Friedjungová[2] and Max van Duijn[1]
[1]*Leiden Institute of Advanced Computer Science, Leiden University, Netherlands*
[2]*Faculty of Information Technology, Czech Technical University in Prague, Czech Republic*

Keywords: Data Visualization, Recommender System, Non-experts, Model.

Abstract: In today's age, there are huge amounts of data being generated every second of every day. Through data visualization, humans can explore, analyse and present it. Choosing a suitable visualization for data is a difficult task, especially for non-experts. Current data visualization recommender systems exist to aid in choosing a visualization, yet suffer from issues such as low accessibility and indecisiveness. The aim of this study is to create a model for a data visualization recommender system for non-experts that resolves these issues. Based on existing work and a survey among data scientists, requirements for a new model were identified and implemented. The result is a question-based model that uses a decision tree and a data visualization classification hierarchy in order to recommend a visualization. Furthermore, it incorporates both task-driven and data characteristics-driven perspectives, whereas existing solutions seem to either convolute these or focus on one of the two exclusively. Based on testing against existing solutions, it is shown that the new model reaches similar results while being simpler, clearer, more versatile, extendable and transparent. The presented model can be applied in the development of new data visualization software or as part of a learning tool.

1 INTRODUCTION

In today's age, there are huge amounts of data being generated every second of every day and Big Data has been one of the hot topics of computer science in recent years. Being the curious species that we are, humans are looking for ways to get the most information out of this vast amount of data that we have available at our fingertips. We are always looking for methods to help us explore, analyze and present it.

A crucial part of this process is data visualization. Data visualization is the representation of information in a visual form, such as a chart, diagram or picture. It can find its place in a variety of areas such as art, marketing, social relations and scientific research. There were over 300 visualization types available at the time of writing this paper (Bostock, 2017). But how do we choose the most suitable one? This is where data visualization recommender systems come in: these systems help with this difficult task that becomes even more difficult when the user is a non-expert.

In this paper we define a 'non-expert user' as someone without professional or specialized knowledge of data visualization. We thus include both complete beginners and users who have general knowledge of data visualization types (e.g. bar charts, pie charts, scatter plots) but have no professional experience in the fields of data science and data communication.

In this study we focus on building a model for a data visualization recommender system aimed at non-expert users. We term our model NEViM: Non-Expert Visualization Model.

Section 2 of this paper places data visualization recommender systems for non-experts in the context of data science. We discuss different types of systems and comment on where the model we are building fits in. Section 3 introduces our research goal and the method we intend to use to fulfill it. Section 4 discusses the results of the work done within our method. We present results of our literature study, existing solutions analysis, survey, model requirements, model construction process and model testing process. We draw conclusions in Section 5 and set an agenda for future work in Section 6.

Kubernátová, P., Friedjungová, M. and Duijn, M.
Knowledge at First Glance: A Model for a Data Visualization Recommender System Suited for Non-expert Users.
DOI: 10.5220/0006851302080219
In *Proceedings of the 7th International Conference on Data Science, Technology and Applications (DATA 2018)*, pages 208-219
ISBN: 978-989-758-318-6

2 CONTEXT

2.1 Data Science

Data science plays an important role in scientific research, as it aids us in collecting, organizing, and interpreting data, so that it can be transformed into valuable knowledge.

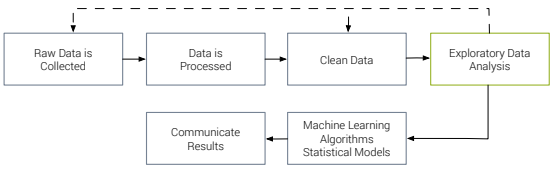

Figure 1: The data science process (O'Neil and Schutt, 2014).

Figure 1 shows a simplified diagram of the data science process. First, real world raw data is collected, processed and cleaned through a process called data munging. Then exploratory data analysis (EDA) follows, during which we might find that we need to collect more data or dedicate more time to cleaning and organizing the current dataset. When finished with EDA, we may use machine learning algorithms, statistical models and data visualization techniques, depending on the type of problem we are trying to solve. Finally, results can be communicated (O'Neil and Schutt, 2014).

Our focus here is on the part of the process concerning exploratory data analysis or EDA. EDA uses a variety of statistical techniques, principles of machine learning, but also, crucially, the data visualization techniques we study in this paper. Please note that data visualization can also be a part of the Communicate Results stage of the data science process (see Figure 1). There is a thin line between data visualizations made for exploration and ones made for explanation, as most exploratory data visualizations also contain some level of explanation and vice-versa.

2.2 Exploratory Data Analysis

Exploratory data analysis (EDA) is not only a critical part of the data science process, it is also a kind of philosophy. You are aiming to understand the data and its shape and connect your understanding of the process that collected the data with the data itself. EDA helps with suggesting hypotheses to test, evaluating the quality of the data, identifying potential need for further collection or cleaning, supporting the selection of appropriate models and techniques and, most importantly for the context of this study, it helps find interesting insights in your data (Tukey, 1970).

2.3 Data Visualization

There are many definitions of the term data visualization. The one used in this study is: data visualization is the representation and presentation of data to facilitate understanding. According to Kirk, our eye and mind are not equipped to easily translate the textual and numeric values of raw data into quantitative and qualitative meaning. "We can look at the data, but we cannot understand it. To truly understand the data, we need to see it in a different kind of form. A visual form." (Kirk, 2016)

Illinsky and Steele describe data visualization as a very powerful tool for identifying patterns, communicating relationships and meaning, inspiring new questions, identifying sub-problems, identifying trends and outliers, discovering or searching for interesting or specific data points (Illinsky and Steele, 2011).

Tamara Munzner made a 3-step model for data visualization design. According to this model, we first need to decide what we want to show. Secondly, we need to motivate why we want to show it. Finally, we need to decide how we are going to show it (Munzner and Maguire, 2015). There are many different types of data visualizations to help us with the third step. However, the challenge remains in choosing the most suitable one. Data visualization recommender systems were made to help with this difficult task. We find that the WHAT and the WHY greatly influence the HOW, thus we aim to build a system that reflects all three aspects of the data visualization design process in some way.

2.4 Data Visualization Recommender Systems

Within this study we define data visualization recommender systems as tools that seek to recommend visualizations which highlight features of interest in data. This definition is based on combining common aspects of definitions in existing work.

While the output of data visualization recommender systems is always a recommendation for data visualization types in some shape or form, the input can differ. It can be, for example, just the data itself, a specification of goals or the specification of aesthetic preferences. The type of input affects the type of recommendation strategy used and consequently the type of the recommender system.

Kaur and Owonibi distinguish 4 types of recommender systems (Kaur and Owonibi, 2017):

- **Data Characteristics Oriented.** These systems recommend visualizations based on data characteristics.

- **Task Oriented.** These systems recommend visualizations based on representational goals as well as data characteristics.
- **Domain Knowledge Oriented.** These systems improve the visualization recommendation process with domain knowledge.
- **User Preferences Oriented.** These systems gather information about the user presentation goals and preferences through user interaction with the visualization system.

The line between different categories of recommendation systems is rather thin and some systems can have ambiguous classifications, as will be discussed below.

3 METHOD

Within this study our aim is to devise a new data visualization recommender system, which is simple and easy to use for non-experts, but can nonetheless compete with existing, often more complex systems. Clearly, we will avoid reinventing the wheel: the current solutions are already good, but we want to see if we can make adjustments that make a system more suitable for non-expert users while maintaining effectiveness (still clearly distinguishing the data visualizations from each other) and performance (recommending the most suitable visualization type).

We will begin by conducting a literature study of previous work done in the field of data visualization recommender systems. We focus on data characteristics-oriented and task-oriented data visualization recommender systems, as this is where our model belongs. The study helps us identify aspects of current solutions which could be utilized in our model and determine which solutions are suitable for the testing of our model.

Next, we run a survey among different data science communities on Facebook and LinkedIn. This way, we ask 88 respondents who have some sort of familiarity with data science and its terminology. The main goals of the survey are to aid us in decisions about our model and, as our model is aimed at non-expert users, to aid us in specifying who exactly these users are.

The findings we make from the literature study, as well as the results of the survey will help us form requirements for our model.

Once we have the requirements, we commence constructing the model. First we choose a suitable base structure. Then we establish the different components of the structure and specify what they will be in our model. Finally, we combine it all together.

We perform two tests on the constructed model. The first test focuses on establishing whether the model is able to produce results similar or identical to existing solutions. The second focuses on testing the extendibility of the model by adding a new type of visualization.

4 RESULTS

4.1 Existing Solutions Study

4.1.1 Data Characteristics Oriented systems

Systems based on data characteristics aim to improve the understanding of the data, of different relationships that exist within the data and of procedures to represent them. Some of the following tools and techniques are not recommendation systems per se but they were a crucial part of the history of this field and foundations for other recommender systems stated, thus we feel it is appropriate to list them as well.

BHARAT

BHARAT was the first system that proposed some rules for determining which type of visualization is appropriate for certain data attributes (Gnanamgari, 1981). As this work was written in 1981, the set of possible visualizations was not as varied as it is today. The system incorporated only the line, pie and bar charts. If the function was continuous, a line chart was recommended. If the user indicated that the range sets could be summed up to a meaningful total, a pie chart was recommended and bar charts were recommended in all the remaining cases. Even though this system would now be considered very basic, it served as the foundation for other systems that followed.

APT

In 1986, Mackinlay proposed to formalize and codify the graphical design specification to automate the graphics generation process (Mackinglay, 1986). His work is based on the work of Joseph Bertin, who, in 1983, came up with a semiology of graphics (Bertin, 1983), where he specified visual variables such as position, size, value, color, orientation etc. and classified them according to which features they communicate best. Mackinlay codified Bertin's semiology into algebraic operators that were used to search for effective presentations of information. He based his findings on the principals of expressiveness and effectiveness. Expressiveness is the idea that graphi-

cal presentations are actually sentences of graphical languages and effectiveness refers to how accurately these presentations are perceived. He would take the encoding technique and formalize it with primitive graphical language (which data visualizations can show this), then he would order these primitive graphical languages using the effectiveness principle.

VizQL(Visual Query Language)

In 2003, Hanrahan revised Mackinlay's specifications into a declarative visual language known as VizQL (Hanrahan, 2006). It is a formal language for describing tables, charts, graphs, maps and time series. The language is capable of translating actions into a database query and then expressing the response graphically.

Tableau and Its Show Me Feature

The introduction of Tableau was a real milestone in the world of data visualization tools. Due to the simple user interface, even inexperienced users could create data visualizations. It was created when Stolte, together with Hanrahan and Chabot, decided to commercialize a system called Polaris (Stolte et al., 2002) under the name Tableau Software. In 2007 Tableau introduced a feature called Show Me (Mackinlay et al., 2007). The Show Me functionality takes advantage of VizQL to automatically present data. At the heart of this feature is a data characteristics-oriented recommendation system. The user selects the data attributes that interest him and Tableau recommends a suitable visualization. Tableau determines the proper visualization type to use by looking at the types of attributes in the data. Each visualization requires specific attribute types to be present before it can be recommended. Furthermore, it also ranks every visualization on familiarity and design best practices. Finally, it recommends the highest-ranked eligible visualization. Mackinlay and his team have also performed interesting user tests with the Show Me feature. They found that the Show Me feature is being used (very) modestly by skilled users (i.e. in only 5.6% of cases).

ManyEyes

Viegas et al. created the first known public website where users could upload data and create interactive visualizations collaboratively: ManyEyes (Viegas et al., 2007). Design choices were made to reflect the effort to find a balance between powerful data-analysis capabilities and accessibility to the non-expert visualization user. The visualizations were created by matching a dataset with one of the 13 types of data visualizations implemented in the tool. They divided the data visualizations into groups by data schemas. A data schema could be, for example, single column textual data. Thus, a bar chart was described as single column textual data and more than one numerical value. The tool closed down in 2015.

Watson Analytics

Since 2014, IBM have been developing a tool called Watson Analytics (IBM, 2017). Watson Analytics uses principles of machine learning and natural language processing to recommend users either questions they can ask about their data, or a specific visualization. However, little is known about how the recommendation system works.

Microsoft Excel's Recommended Charts Feature

In the 2013 release of Microsoft Excel, a new feature called Recommended Charts was introduced. The user can select the data they want to visualize and Excel recommends a suitable visualization (Microsoft, 2017). However, Microsoft does not share exactly how this process is carried out, making it less suitable as a source of inspiration.

SEEDB

In 2015 Vartak et al. proposed an engine called SEEDB (Vartak et al., 2015). They judge the interestingness of a visualization based on the following theory: a visualization is likely to be interesting if it displays large deviations from some reference (e.g. another dataset, historical data, or the rest of the data). This helps them identify the most interesting visualizations from a large set of potential visualizations. They identified that there are more aspects that determine the interestingness of a visualization, such as aesthetics, user preference, metadata and user tasks. A full-fledged visualization recommendation system should take into account a combination of these aspects. A major disadvantage of SEEDB is that it only uses variations of bar charts and line charts. As far as we know SEEDB was never deployed.

Voyager

In 2016, Wongsuphasawat et al. developed a visualization recommendation web application called Voyager (Wongsuphasawat et al., 2016), based on the Compass recommendation engine (Wongsuphasawat, 2017) and a high-level specification language called Vega-lite (Satyanarayan et al., 2017). It couples browsing with visualization recommendation to support exploration of multivariate, tabular data.

Google Sheets and Its Explore Feature

Google Sheets (Google, 2017) is a tool which allows users to create, edit and share spreadsheets. It was introduced in 2007 and is very similar to Microsoft Excel. In June of 2017, the tool was extended with the Explore Feature, which helps with automatic chart building and data visualization. It uses elements of artificial intelligence and natural language processing to recommend users questions they might want to ask about their data, as well as recommending data visualizations that best suit their data. In the documentation for this feature, Google specifies each of the included data visualizations using functions and conditions that have to be fulfilled in order for that particular data visualization to be recommended. However, a couple of visualizations have the same conditions and it is not revealed how the most suitable data visualization is chosen.

4.1.2 Task Oriented Systems

Task-oriented systems aim to design different techniques to infer the representational goal or a user's intentions. In 1990 Roth and Mattis were the first to identify different domain-independent information seeking goals, such as comparison, distribution, correlation etc. (Roth and Mattis, 1990). Also in 1990, Wehrend and Lewis proposed a classification scheme based on sets of representational goals (Wehrend and Lewis, 1990). It was in the form of a 2D matrix where the columns were data attributes, the rows representational goals and the cells data visualizations. To find a visualization, the user had to divide the problem into subproblems, until for each subproblem it was possible to find an entry in the matrix. A representation for the original complex problem could then be found by combining the candidate representation methods for the subproblems. Unfortunately, the complete matrix was not published so it is unknown which specific types of data visualizations were included.

IMPROVISE

In the previous studies, the user task list was manually created. However, in 1998, Zhou and Feiner introduced advanced linguistic techniques to automate the derivation of the user task from a natural language query (Zhou and Feiner, 1998). They introduced a visual task taxonomy to automate the process of gaining presentation intents from the text. For example, the visual task Focus implies that visual techniques such as Enlarge or Highlight could be used. This taxonomy is implemented in IMPROVISE. Zhou and Feiner show how IMPROVISE generates a visual narrative from speech to present an overview of a hospital patient's information to a nurse. To achieve this goal, it constructs a structure diagram that organizes various information (e.g. IV lines) around a core component (the patient's body). In a top-down design manner, IMPROVISE first creates an 'empty' structure diagram and then populates it with components by partitioning and encoding the patient information into different groups.

HARVEST

In 2009 Gotz and Wen introduced a novel behavior-driven approach (Gotz and Wen, 2009). Instead of needing explicit task descriptions, they use implicit task information obtained by monitoring users' behavior to make recommendation more effective. The Behavior-Driven Visualization Recommendation (BVDR) approach has two phases. In the first phase of BDVR, they detect four predefined patterns from user activity. In the second phase, they feed the detected patterns into a recommendation algorithm, which infers user intent in terms of common visual tasks (e.g. comparison) and suggests visualizations that better support the user's needs. The inferred visual task is used together with the properties of the data to retrieve a list of potentially useful visual metaphors from a visualization example corpus made by Zhou and Chen (Zhou et al., 2002). It contains over 300 examples from a wide variety of sources. Unfortunately, we were not able to access this corpus.

All in all, we identify some pitfalls of the existing systems. Such as them not being accessible enough, too complicated, too formal and too secretive when it comes to their recommendation process. The biggest pitfall is that the result of their recommendation process is most commonly a set of data visualizations, which, in our opinion, leaves the users a bit further than they started, but still nowhere, because they still have to choose the most suitable visualization. The possibilities have been narrowed, but a decision still must be made. We hope to avoid these pitfalls within our model. We establish that we are going to test our model against the solutions available to us. This means Tableau, Watson Analytics, Excel, Voyager and Google Sheets. Please note that we are going to compare against the recommendation system features of the tools, not the tools as a whole.

4.2 Exploratory Survey

We run a survey among different data science communities on Facebook and LinkedIn. This way, we get respondents who have some sort of familiarity with data science and its terminology. The main goals of the survey are to aid us in making decisions about our

model and specifying the term non-expert user.

4.2.1 Participants

In total, we gathered 88 valid responses (*n=88*). Out of the 88 respondents, 78% (*n=69*) were male and 22% (*n=19*) female. The average age was 29.86 years.

We had asked the respondents to indicate their knowledge level on a scale of 1 to 10, 1 being beginner and 10 being expert. The average knowledge level was 5.70. We opted to divide the scale into three ranges in the following way: 1-3 are beginners, 4-7 are non-experts and 8-10 are experts. According to our ranges we had 26% (*n=23*) beginner level, 44% non-expert (*n=39*) level and 30% (*n=26*) expert level respondents.

4.2.2 Results

We make the following findings from the results of our survey:

- For all groups, the main purpose of making data visualizations was for analysis (65% of beginners, 64% of non-experts, 58% of experts).
- All types of users choose data visualizations mainly according to: the characteristics of their data (57% of beginners, 62% of non-experts, 65% of experts) and the tasks that they want to perform (48% of beginners, 51% of non-experts, 62% of experts).
- For all groups, the two most used visualizations are bar charts (17% of beginners, 38% of non-experts, 35% of experts) and scatter plots (43% of beginners, 26% of non-experts, 31% of experts).
- All groups were mostly unable to name an existing data visualization recommendation system (0% able vs. 100% unable for beginners, 5% able vs. 95% unable for non-experts and 4% able vs. 96% unable for experts).
- All groups would be willing to use a data visualization recommendation system, although experts were less willing than beginners and non-experts (100% willing vs. 0% not willing for beginners, 87% willing vs. 13% not willing for non-experts and 77% willing vs. 23% not willing for experts).

To summarize, we have learned that non-experts make data visualizations mainly for the purpose of analysis. When they select a suitable data visualization type, they do so according to the characteristics of their data and the tasks they want to perform. Their most used visualization types are bar charts and scatter plots. They are not familiar with data visualization recommender systems but are mostly willing to use one. We also learned that there is not much difference between the approaches of beginners, non-expert and expert users, which was unexpected.

4.3 Model Requirements

Based on research of previous approaches to our problem and the results of our survey, we have identified the following requirements which NEViM should fulfill:

1. **Simplicity** - The model should be simple enough to be used by non-experts. It must have good flow and a very straightforward base structure.
2. **Clarity** - We aim for the result of our recommendation system to be one data visualization. Not a set, like in some current tools. This means that the underlying classification hierarchy of data visualizations must be clear and unambiguous.
3. **Versatility** - We want our model to combine different kinds of recommendation systems. From our survey we learn that when users select a suitable data visualization type, they do so based on the characteristics of their data and the tasks they want to perform. Based on this we incorporate a data characteristics-oriented and task-oriented approach. Furthermore, we want our model to be easily implemented in different programming languages and environments.
4. **Extensibility** - Our aim is for our model to be easily extendable. We want the process of adding visualizations into the model to be simple. We want it to be a useful skeleton which can be easily extended to include automatic visualizations etc.
5. **Education** - We want our model to not only function as a recommender system, but also as a learning tool.
6. **Transparency** - Once we recommend a visualization, we want the users to see, why the particular visualization was recommended, meaning that the path to a visualization recommendation through our model has to be retraceable.
7. **Self-learning** - We want our model to be able to improve itself. This means, amongst other things, that it should be machine learning friendly.
8. **Competitiveness** - We want our model to still produce results which are comparable to results from other systems.

4.4 Constructing NEViM

4.4.1 What Base Structure to Use for NEViM?

Since the aim of our model is to help a user *decide* which data visualization to use, the obvious choice seemed to be the structure of decision trees. A decision tree has four main parts: a root node, internal nodes, leaf nodes and branches. The biggest advantages of decision trees are that they can help uncover unknown alternative solutions to a problem and that they are well suited for machine learning methods.

Once we determined that the decision tree was a possible base structure, we needed to specify what our root node, internal nodes, leaf nodes and branches would be. It was clear that the leaf nodes would be the different types of data visualizations since that was the outcome that we wanted to achieve. The root node, internal nodes and branches are inspired by Akinator, the Web Genie. Akinator is a game that attempts to determine which character the player is thinking of by asking a series of questions. The structure hidden under the user interface is a decision tree, as in the case of NEViM.

Our model's root and internal nodes are questions which possess the ability to clearly distinguish different types of data visualizations. The branches are 'yes' or 'no' answers to those questions.

4.4.2 What Questions to Ask? (Establishing the Internal Nodes and Root Node)

The biggest challenge in constructing questions for our model was that they must be understandable for non-experts, yet every question should get the user closer to a data visualization recommendation. This means that the subjects of the questions must be features that distinguish the different data visualizations from each other. The key to solving this problem is to base the questions on a clear classification hierarchy. As far as we know, there is no one specific classification hierarchy of data visualizations which would be used globally. We researched different methods of classification and combined them together to derive a classification of our own. This was a very time-consuming process. We went through a total of 19 books (O'Neil and Schutt, 2014; Kirk, 2016; Illinsky and Steele, 2011; Munzner and Maguire, 2015; Gnanamgari, 1981; Evergreen, 2016; Yau, 2011; Yau, 2013; Heer et al., 2010; Hardin et al., 2012; Yuk and Diamond, 2014; Brath and Jonker, 2015; Brner and Polley, 2014; Telea, 2007; Brner, 2015; Ware, 2010; Ware, 2012; Stacey et al., 2015; Hinderman, 2015) and for each one, we constructed a diagram showing the classification that was described in the text.

We examined the classification hierarchies from books together with hierarchies available from web resources and existing tools. We also made note of any advantages or disadvantages of a specific data visualization, if they were listed. For example in several sources (O'Neil and Schutt, 2014; Kirk, 2016; Illinsky and Steele, 2011) the authors stated that the pie chart is not suitable for when you have more than 7 parts. The advantages and disadvantages reflected features of the data visualizations that could determine whether they are candidates for recommendation or not, so they are crucial for the final model.

We identified that there are two basic views that the classifications incorporate. The first one is a view from the perspective of the task the user wants to perform. The second is a view from the perspective of the characteristics of the data the user has available. This is in line with data characteristics and task oriented recommendation systems (Kaur and Owonibi, 2017).

We have identified a prominent issue in the classification hierarchies: they mix different views into one without making a clear distinction between them. To avoid this issue, we have selected the root node of our model to be a question which would distinguish between two views. The first view is from a task-based perspective and it uses the representational goal or user's intentions behind visualizing the data to recommend a suitable visualization. The second view is from a data-driven perspective, where a visualization recommendation is made based on gathering information about the user's data. The root node of NEViM is a question asking "Do you know what your main task is?" If the user answers "Yes", he is taken in to the task-based branch. If he answers "No", he is taken straight into the data characteristics-based branch.

Once we established the root node, we had to come up with internal nodes. The internal nodes are questions which possess the ability to clearly distinguish different types of data visualizations. The subject of such a question must be something that we define as a distinguishing feature. Based on the findings we made in previous paragraphs, we have established a list of distinguishing features and their hierarchy.

Based on the distinguishing features, we have constructed questions that ask whether that feature is present or not. You can see an example of such questions in Figure 3.

4.4.3 What Data Visualizations to Include? (Establishing the Leaf Nodes)

Once we had figured out our model's base structure, distinguishing features and questions, the challenge was, which data visualizations to include. We knew that we would not be able to cover all the 300 types

of data visualizations available (Bostock, 2017) in the initial version of our model. We took a rather quantitative approach to the problem. We went through all the different classification hierarchies we constructed previously and extracted a list of the data visualizations that occur. We removed duplicates (different names for the same visualization, different layouts of the same visualization) and we counted how many times each data visualization occurred. The ones that occurred 5 times or more were included in our final model. The final list contains 29 data visualizations and you can see it below. Since one of our requirements for the final model is easy extensibility, we feel that 29 data visualizations are appropriate for the initial model.

Table 1: Data visualizations included in NEViM.

Bar Chart	Bubble Chart	Cartogram
Choropleth Map	Clustered Bar	Connected Dot
Connection Map	Density Plot	Dot Map
Flow Map	Heat Map	Histogram
Line Chart	Network	Pie Chart
Proportional Map	Radar Plot	Scatter Plot
SPLOM	Slope Graph	Small Multiples
Stacked Area	Stacked Bar	Stacked Line
Table	Timeline	Treemap
Parallel Coordinates		

4.4.4 Putting It All Together

We classified each of our leaf nodes (data visualizations) using the distinguishing features we constructed previously. This revealed the path of internal nodes and branches that leads to a certain leaf node. In other words, it revealed which questions have to be answered and how in order to get to a certain data visualization.

We then combined all the classifications together to construct the final model[1]. The model has 107 internal nodes and 105 leaf nodes. The model always results in a recommendation. If no other suitable visualization is found, we recommend to use a table by default. Tableau does this as well.

4.5 Testing the Model

4.5.1 Can the Model Compete with Existing Solutions?

We carried out tests to determine whether our model was able to compete with existing systems in terms of similarity of solutions. We obtained 10 different test data sets with various features (See Table 2). The data sets were preprocessed to remove invalid entries and to ensure that all the attributes were of the correct data type.

For each data set, we formulated an example question that a potential user is aiming to answer. This was done in order to determine which attributes of the data would be used in the recommendation procedure. Most existing tools require the user to select the specific attributes that they want to use for their data visualization. By specifying these for each data set we attempt to mimic this behavior. Table 2 shows the data sets along with their descriptions.

We tested our model against existing solutions which are freely available: Tableau (10.1.1), Watson Analytics (version available in July 2017), Microsoft Excel (15.28 Mac), Voyager (2) and Google Sheets (version available in July 2017). For each system and every data set, we aimed to achieve a recommendation for a data visualization that would answer the question and incorporate all the specified attributes in one graph as there is no possible way to answer the question without incorporating the specified attributes. Some systems solve more complex questions by creating a series of different data visualizations, with each visualization incorporating a different combination of attributes. We excluded such solutions from our test results because we feel that it is a workaround. For Microsoft Excel and Google Sheets, the recommendation process results in several recommendations and the systems do not rank them. For these cases we recorded all valid recommendations.

Results

For data set 1, all systems recommended a bar chart. Excel and Google Sheets also recommended a pie chart. The recommendations for data set 2 were either line charts or bar charts. The specified question could be answered by either of these. Watson Analytics was not able to give a recommendation because it could not recognize that the average price attribute was a number. We have attempted resolving this issue but were not able to. For data set 3, the majority recommendation was a clustered bar chart, in line with the recommendation made by NEViM. Data set 4 proved to be challenging for Voyager and Watson Analytics. Since the data was hierarchical and the question was asking to see parts-of-whole, a suitable solution would be a tree map. A pie chart shows parts-of-whole, but does not indicate hierarchy. The question asked for data set 5 could be answered using different types of data visualizations. Since it is asking to analyze the correlation between 2 variables, a scatter plot is a suitable solution. All systems recommended it. Data set number 6 was an example of a social net-

[1]The whole model as well as a prototype can be viewed at a website dedicated to this research project: http://www.datavisguide.com

Table 2: Results of the competitiveness test.

Data set	Description	Records	Question	Used attributes	Excel	Google Sheets	Tableau	Voyager	Watson Analytics	NEViM
1	Favourite subjects within a class of students	7	What does the composition of the data look like?	subject, no. of students	bar chart, pie chart	bar chart, pie chart	bar chart	bar chart	bar chart	bar chart
2	Average prices of cigarettes over several years	8	What was the development of the cigarette price over the years?	year, average price	line chart, bar chart	line chart	line chart	bar chart	none	line chart
3	Percentage of men and women in EU countries for 2016	28	Which 5 countries have the highest percentage of females?	country, % of men, % of women	clustered bar chart, scatter plot, stacked bar chart	clustered bar chart	proportional symbol map	scatter plot	clustered bar chart	clustered bar chart
4	Causes of death in Kenya in 2012	12	How big of a part does each cause take?	cause of death, no. of deaths, % of total	none	pie chart	tree map	none	none	tree map
5	Daily ice cream sales information with temperature	30	Are ice cream sales related to the weather?	income, temperature	scatter plot, clustered bar chart, line chart, stacked bar chart	line chart, scatter plot, clustered bar chart	scatter plot	scatter plot	scatter plot	scatter plot
6	Email communication between researchers working together	461	Which researcher is connected to most people?	sender, receiver	none	none	none	scatter plot	none	network
7	Finishing times of runners in the 2014 Boston Marathon	32K	Which finishing time interval was the most common?	finishing time	scatter plot, line chart	line chart, histogram	histogram	none	histogram	histogram
8	Records of UFO sightings with detailed information	80K	Are there any clusters of locations where UFOs have been seen more often?	latitude, longitude	none	none	none	none	none	dot map
9	List of cars and their parameters	393	Are there any relationships between the different parameters?	miles per gallon, no. of cylinders, displacement, horsepower, weight, acceleration, year	stacked line chart	none	none	none	none	parallel coordinates
10	Origins and destinations of flights within the US	4K	Which city has the most ingoing and outgoing flights?	flight origin, flight destination	none	none	proportional symbol map	none	none	connection map

work, thus the most suitable visualization would be a network. However, the answer to the specified question could also be answered with a scatter plot as suggested by Voyager. This is because networks can also be represented as adjacency matrices and the scatter plot generated by Voyager is essentially an adjacency matrix. Data set 7 and its question were aimed at visualizing distributions. Distributions can be visualized, among others with histograms, scatter plots and line charts. Data set 8 was an example of spatial data. Spatial data is best visualized through maps. Tableau offers map visualizations but we suspect that it cannot plot on the map according to latitude and longitude coordinates. Watson Analytics and Google Sheets have the same issue. Microsoft Excel and Voyager do not support maps at all. In Data set 9 the answer to the question was revealed through comparing 7 attributes. This meant that the visualization has to support 7 different variables. Both stacked line chart and parallel coordinates are valid solutions. The final data set 10 was again spatial. This time it could be solved through plotting on a map but also by analyzing the distribution of the data set. Both proportional symbol map and connection map (as a flight implies a connection between two cities) are valid solutions.

Overall, we can observe that NEViM provided usable solutions in all cases. The users have several paths that they can take through NEViM to get to a recommendation, depending on what information they know about their data or their task. NEViM has an advantage that it is not limited by implementation. Since two of our data sets were aimed at spatial data visualization (9 and 10) and one at network data visualization (6), some systems were not able to make recommendations simply because they do not support such visualization types. Furthermore, NEViM includes more types of visualizations than any of the current systems, which results in recommendations for specialty visualizations that can be more suitable for a certain task. Another advantage is that it always results in only one recommendation, unlike Microsoft Excel or Google Sheets, where the user has to choose which one out of the set of recommendations to use. According to our survey, the most used visualization tool which incorporates a recommender system is Tableau (28% of non-expert respondents). From the result table, we can see that in 5 out of 7 valid cases, NEViM made the same recommendation as Tableau.

Furthermore, in data set 3 Tableau also made a recommendation for a Clustered Bar Chart, like NEViM did, but it was not the resulting recommendation. One of the attributes was the name of a country, so Tableau evaluated the data as spatial. We have noticed that whenever there is a geographical attribute, Tableau prefers to recommend maps, even though they might not be the most suitable solution.

4.5.2 Adding a New Data Visualization

We demonstrate that our model is easily extensible by showing the process of adding a new data visualization type - a Sankey diagram. Sankey diagrams are specific types of flow diagrams and they display quantities in proportion to one another. An example of a Sankey diagram can be seen in Figure 2.

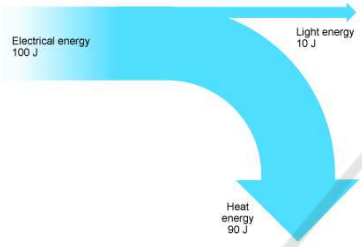

Figure 2: Example of a Sankey diagram showing the distribution of energy in a filament lamp (BBC, 2016).

We look into the classifications that we already have and search for the most similar one. We find out that the Tree Map has the same classification. So we need to find a distinguishing feature between a Tree Map and a Sankey diagram. That feature is, that a Sankey diagram shows flow. We search through the model and find occurrences of a Tree Map. We then add a question asking "Do you want to show flow?". If the user answers "Yes", he gets a recommendation for a Sankey diagram. If he answers "No" he gets a Tree Map. Figure 3 shows the two paths that a user of NEViM can take to get to the Sankey diagram.

5 DISCUSSION & CONCLUSIONS

We managed to build a model for a data visualization recommender system suited to non-experts called NEViM. Through testing, we have managed to show that the resulting recommendations are similar or identical to the ones generated by existing solutions. Based on a review of existing work and a exploratory survey among users, we have put together requirements. This is a short evaluation of how NEViM managed to fulfill these:

1. **Simplicity** - Thanks to its question-based structure, using the model is simple. The user only has to answer yes or no questions. The basic structure is very straightforward.
2. **Clarity** - The result of our recommendation system is a single data visualization, making it very clear. We believe that non-expert users need a clear answer to their visualization problem. If they are given a choice between two or more visualizations in the end, we believe that we have failed at the task of recommending them the most suitable one. We have narrowed their choices, but still have not provided a clear answer. However, this decision seems to be a controversial one, so it definitely needs to be validated through a user study (See Section 6.) In the case that none of the data visualizations within the model are determined as suitable, the model still makes a recommendation to visualize using a table.
3. **Versatility** - NEViM combines two different types of data visualization recommendation systems as defined in (Kaur and Owonibi, 2017): task-oriented and data characteristics-oriented. These two types are distinguished by two different starting points within our model. Thanks to its base structure the model can be easily implemented in various different programming languages and environments.
4. **Extensibility** - To illustrate the extensibility of the model, we have added the Sankey diagram visualization. This proved to be a doable task.
5. **Education** - This requirement has not been met yet. For suggestions on how we mean to fulfill it, see Section 6.
6. **Transparency** - The traversal through our model is logical enough that it is clear why a certain type of data visualization was recommended.
7. **Self-learning** - Our model is machine learning friendly and techniques can be applied for it to be able to self-learn. See our Section 6.
8. **Competitiveness** - Through testing we have proved that our model produces recommendations similar or identical to existing solutions. It provided suitable solutions for all cases tested, unlike existing solutions.

A possible disadvantage of NEViM could be that the user has to either know what their main task is, or know what type of data they have. The question is, whether non-expert users will be able to determine this. We believe that this could be fixed through user testing to validate the overall structure of the model

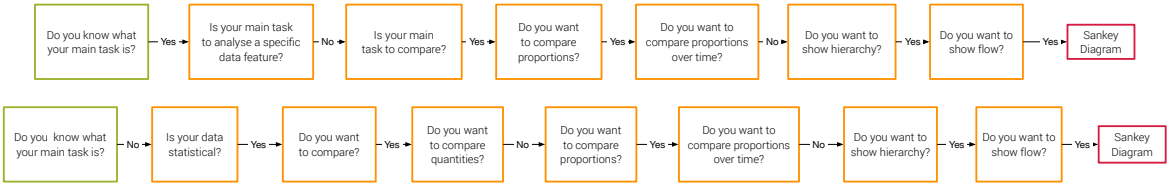

Figure 3: Two possible paths to reach a Sankey diagram (left: task-based, right: data-based).

as well as the quality of the questions. The questions could be checked by a linguistics expert to see whether the wording is suitable and does not lead to possible ambiguous interpretations.

Another disadvantage might lie in the fact that since we use data science terminology in our questions, we risk that non-experts might not be familiar with it and might not be able to answer the question. A solution could be to clarify the terms using a dictionary definition, which could pop up when the user hovers over the unfamiliar term. The solution is more part of the implementation phase, not the theoretical phase which we discuss here.

A difficulty in the usability of our model might be that the traversal through it is quite lengthy. This is due to the chosen question-based approach. A potential fix for this could be to present some parts of the model in the form of a multiple choice question. This way, the user could see beforehand what other options are available and might find a more suitable task they want to perform. This is once again a problem that could be fixed easier in the implementation phase.

We have questioned whether the choice to recommend a table when no other suitable visualization is found is the correct one. There is an ongoing debate about when it is best to not visualize things, as discussed by Stephanie Evergreen (Evergreen, 2016). Within the implementation phase, data could be collected to find out in how many cases the Table option is reached, to identify whether it is necessary to further address this issue.

6 FUTURE WORK

We have proved that there is definitely a place for our model in the data science world. The logical next step would be to perform more tests with more data sets and make improvements to the model. Then the model could be tested with non-expert users. Such a user study could evaluate the usability of the model as well as its contribution.

The model could be implemented as a web application and users could rate the resulting recommendations, suggest new paths through the model or request new visualization types to be included. This would also validate the question paths that we have designed. The final recommendation could be enhanced with useful information about the data visualization type, tips on how to construct it, which tools to use and examples of already made instances. This would transform the model into a very useful educative tool and fulfill the Education requirement that we have set.

Another possible extension to the model could be to add another view which would incorporate information about the domain that the user's data comes from. There are data visualizations that are more suited for a specific data domain than others. For example, the area of economics has special types of data visualizations that are more suited to exposing different economic indicators. This would make the model part of the domain knowledge oriented data visualization systems recommender systems category according to (Kaur and Owonibi, 2017).

Thanks to its structure, NEViM is machine learning friendly. For example, neural networks could be used to make the model self-learning and self-improving.

We could introduce different features that could influence the visualization ranking - e.g. perceptual qualities of different data visualization types. Now that we have established a successful base, the possibilities for further development are endless.

ACKNOWLEDGEMENTS

Research supported by SGS grant No. SGS17/210/OHK3/3T/18 and GACR grant No. GA18-18080S.

REFERENCES

BBC (2016). Heat transfer and efficiency.

Bertin, J. (1983). *Semiology of graphics: diagrams, networks, maps*.

Bostock, M. (2017). Data-driven documents.

Brath, R. and Jonker, D. (2015). *Graph analysis and visualization: discovering business opportunity in linked data*. John Wiley & Sons, Hoboken, NJ.

Brner, K. (2015). *Atlas of knowledge: Anyone can map*. MIT Press, Cambridge, MA.

Brner, K. and Polley, D. E. (2014). *Visual insights: A practical guide to making sense of data*. MIT Press, Cambridge, MA.

Evergreen, S. D. (2016). *Effective data visualization: The right chart for your data*. SAGE Publications, Thousand Oaks, CA.

Gnanamgari, S. (1981). *Information presentation through default displays*. PhD thesis, Univ. of Pennsylvania, Philadelphia, PA.

Google (2017). Chart and graph types.

Gotz, D. and Wen, Z. (2009). Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, New York, NY.

Hanrahan, P. (2006). Vizql: a language for query, analysis and visualization. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, New York, NY.

Hardin, M. et al. (2012). Which chart or graph is right for you?. tell impactful stories with data. *Tableau Software*.

Heer, J. et al. (2010). A tour through the visualization zoo. *Queue*, 8.5.

Hinderman, B. (2015). *Building responsive data visualization for the web*. John Wiley & Sons, Hoboken, NJ.

IBM (2017). Smart data analysis and visualization.

Illinsky, N. and Steele, J. (2011). *Designing data visualizations: representing informational relationships*. O'Reilly Media, Sebastopol, CA.

Kaur, P. and Owonibi, M. (2017). A review on visualization recommendation strategies. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 266–273, Porto, Portugal.

Kirk, A. (2016). *Data visualization: A handbook for data driven design*. SAGE, London,UK.

Mackinglay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5.2:110–141.

Mackinlay, J. et al. (2007). Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13.6.

Microsoft (2017). Available chart types in office.

Munzner, T. and Maguire, E. (2015). *Visualization analysis and design*. CRC Press, Boca Raton, FL.

O'Neil, C. and Schutt, R. (2014). *Doing Data Science: Straight Talk From The Frontline*. OReilly Media, Sebastopol,CA.

Roth, S. F. and Mattis, J. (1990). Data characterization for intelligent graphics presentation. *SIGCHI Conference on Human Factors in Computing Systems*.

Satyanarayan, A. et al. (2017). Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23.1:341–350.

Stacey, M. et al. (2015). *Visual intelligence: Microsoft tools and techniques for visualizing data*. John Wiley & Sons, Hoboken, NJ.

Stolte, C. et al. (2002). Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8.1:52–65.

Telea, A. C. (2007). *Data visualization: principles and practice*. CRC Press, Boca Raton, FL.

Tukey, J. W. (1970). *Exploratory Data Analysis*. Addison-Wesley, Reading,MA.

Vartak, M. et al. (2015). Seedb: supporting visual analytics with data-driven recommendations. *VLDB*.

Viegas, F. et al. (2007). Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13.6.

Ware, C. (2010). *Visual thinking: For design*. Morgan Kaufmann, Burlington, MA.

Ware, C. (2012). *Information visualization: perception for design*. Elsevier, Amsterdam, NL.

Wehrend, S. and Lewis, C. (1990). A problem-oriented classification of visualization techniques. In *Proceedings of the 1st Conference on Visualization'90*, Los Alamitos, CA.

Wongsuphasawat, K. (2017). Vega compass.

Wongsuphasawat, K. et al. (2016). Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22.1:649–658.

Yau, N. (2011). *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. John Wiley and Sons, Hoboken, NJ.

Yau, N. (2013). *Data points: Visualization that means something*. John Wiley & Sons, Hoboken, NJ.

Yuk, M. and Diamond, S. (2014). *Data visualization for dummies*. John Wiley & Sons, Hoboken, NJ.

Zhou, M. X. et al. (2002). Building a visual database for example-based graphics generation. *INFOVIS 2002 IEEE Symposium*.

Zhou, M. X. and Feiner, S. K. (1998). Visual task characterization for automated visual discourse synthesis. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, Boston, MA.