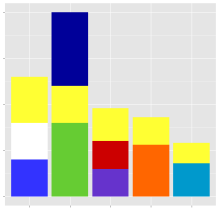
Simpsonyzer

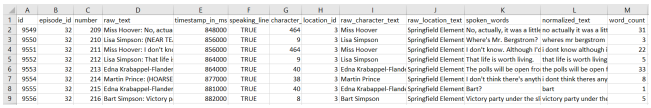


Oleksandr Shabelnyk* Ivo Vidovic† Bernhard Bachinger‡

1 Motivation

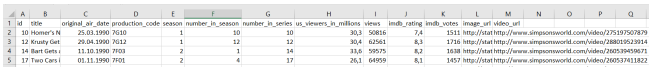
1.1 Dataset

We are going to visualize a dataset consisting of four csv files with various information on all episodes of the popular TV-series The Simpsons, seasons 1 to 26. The data originates from the following webpage: https://www.kaggle.com/wcukierski/the-simpsons-by-the-data . According to this website inspiration and credit for gathering the data goes to Todd Schneider. These csv-files contain for example every script line of every episode. Furthermore, corresponding locations and characters are listed as well. We took screenshot of two biggest files with their headers and couple of rows which can be seen on figure 1 and 2 below.



id	episode_id	number	raw_text	timestamp_in_ms	speaking_line	character_id	location_id	raw_character_text	raw_location_text	spoken_words	normalized_text	word_count
9549	32	209	Miss Hoover: No, actua	848000	TRUE	464	3	Miss Hoover	Springfield Element	No, actually, it was a little	no actually it was a litt	31
9550	32	210	Lisa Simpson: (NEAR TE	856000	TRUE	9	3	Lisa Simpson	Springfield Element	Where's Mr. Bergstrom?	wheres mr bergstrom	3
9551	32	211	Miss Hoover: I don't kn	856000	TRUE	464	3	Miss Hoover	Springfield Element	I don't know. Although I'd i	dont know although i	22
9552	32	212	Lisa Simpson: That life i	864000	TRUE	9	3	Lisa Simpson	Springfield Element	That life is worth living.	that life is worth living	5
9553	32	213	Edna Krabappel-Flander	864000	TRUE	40	3	Edna Krabappel-Flands	Springfield Element	The polls will be open fro	the polls will be open f	33
9554	32	214	Martin Prince: (HOARSE	877000	TRUE	38	3	Martin Prince	Springfield Element	I don't think there's anyth	i dont think theres any	8
9555	32	215	Edna Krabappel-Flander	881000	TRUE	40	3	Edna Krabappel-Flands	Springfield Element	Bart?	bart	1
9556	32	216	Bart Simpson: Victory p	882000	TRUE	8	3	Bart Simpson	Springfield Element	Victory party under the sl	victory party under the	5

Figure 1: Raw data excerpt



id	title	original_air_date	production_code	season	number_in_season	number_in_series	us_viewers_in_millions	views	imdb_rating	imdb_votes	image_url	video_url
10	Homer's N	25.03.1990	7G10	1	10	10	30,3	50816	7,4	1511	http://stat.ht	http://www.simpsonsworld.com/video/275197507879
12	Krusty Get	29.04.1990	7G12	1	12	12	30,4	62561	8,3	1716	http://stat.ht	http://www.simpsonsworld.com/video/288019523914
14	Bart Gets i	11.10.1990	7F01	2	1	14	33,6	59575	8,2	1638	http://stat.ht	http://www.simpsonsworld.com/video/260539459671
17	Two Cars i	01.11.1990	7F01	2	4	17	26,1	64959	8,1	1457	http://stat.ht	http://www.simpsonsworld.com/video/260537411822

Figure 2: Raw data excerpt

1.2 Methodology

To develop our project, we have decided to apply design study technique. Since we would like to combine technical implementation and user experience evaluation, we decided that this method will suit our needs at most. Moreover, as per the Multi-Level Typology of Abstract Visualization Tasks introduced by Matthew Brehmer and Tamara Munzner the why part of our visualization would be categorized as enjoy. [1]

1.3 Target Group

Our target group are the fans of the TV-series The Simpsons. We want to make a tool which will allow them to discover facts about the Simpsons they may have not known about while they are enjoying exploring the data. After they used our visualization they can brag about their great knowledge to their friends.

1.4 User tasks

Main tasks:

- Find out information about a specific episode or season
- Find out interesting facts about the favorite character
- Determine how various aspects of the series changed over the time
- Find out what are the best rated episodes

Secondary tasks derived from the main ones:

- Who talks the most in a season or episode.
- To whom a character talks most frequently.
- Who said a word how many times.
- Where does the character speak most of the time.
- Determine which characters had more spoken words at what time.

Some of introduced problems, e.g. tasks, cannot be solved with standard tools, like for example the second task: ”Find out interesting facts about the favorite character”. It needs to be approached with help of sophisticated algorithms and the data needs to be preprocessed. Moreover, we are going to implement interaction and filters which will make our visualization even more unique.

2 Related Work

Because of the high popularity of the Simpsons there is quite a high number of related materials, but as far as visualization is concerned not so many. Below we introduce a couple of examples.

”The Simpsons Viszpedia” (https://public.tableau.com/en-us/s/gallery/visualizing-simpsons) is a Tableau dashboard which incorporates some similar elements to our project, for instance the barchart with ratings. However the creators used instead of the IMDB ratings the number of viewers. In regard to our project showing IMDB ratings is more preferable, because of our target group, so we used instead the mentioned above IMDB ratings. Another example of similarity to our tool would be the tab ”Who is the main character?”. It shows how often a character is featured in plot. Simpsonyzer can show this information also, by every episode if needed, but on top of this other characters the selected character speaks to most frequently is shown as well. This way the user can not only know who is the main character in a particular episode/season, but also get more useful information.

*e-mail:oleksandr.shabelnyk@univie.ac.at
†e-mail:******@unet.univie.ac.at
‡e-mail:a1368644@unet.univie.ac.at

Todd W. Schneider http://toddwschneider.com/posts/the-simpsons-by-the-data/ present a set of views with mostly barcharts, but they are only static. Our tool is very dynamic, interactive and adaptive. Anyway, if not considering the lack of interactivity, Todd Schneider's work contains similar barcharts to those in our tool, even if their objective differs from the Simpsonyzer's one. An good example for ti would be the amount of spoken words. In our project they are used to determent how prominent a character is in which episode. In the article the author wants to say this way only what characters are the most "important" one. However, it is rather wrong to compare this work to the Simsonyzer, since Todd Schneider's intention was to tell a specific story. Our tool offers its users to create and tell others their own stories.

3 Approach

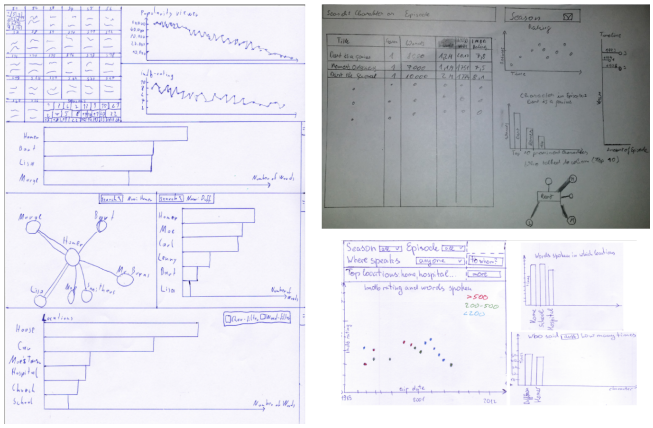


Figure 3: Lo - Fi prototypes, drawn by hand.

After determining user tasks and target group introduced previously we began designing our visualization tool by drawing low fidelity prototypes [1]. Based on users' tasks, we came up with three dashboards each incorporating completely different approach. 3 Later we tested them by following criteria:

- User tasks fulfillment
- Applicability to the target audience
- Usability

We created couple of use case scenarios and tested our prototypes based on them. After the tests we had to combine our three dashboard into only one. Since it is almost impossible to create the perfect dashboard, we considered what elements of the three dashboards would be most suitable for the best fulfillment of the tasks and the best user experience, based on our target audience. Furthermore, we performed the same steps for Hi-Fi prototype [2]. On fig. 4 you can see what our Hi-Fi looked like.

Starting from section 3.2 we will be discussing each element/view we finally came up with more detailed.

3.1 Theoretical background

Firstly we would like to introduce two terms we will be often referring to: marks and channels. Tamara Munzner defines a mark as a basic primitive graphical element in the image. [2, p. 90] An example for a mark would be a point, line or area. A channel controls the appearance of the marks, "independent of the dimensionality of the geometric primitive". [2, p. 91] For instance, a channel is the way marks are positioned, colored, shaped, titled, etc.

[1] Further denoted as milestone 2 or simply M2

[2] Further denoted as milestone 3 or simply M3

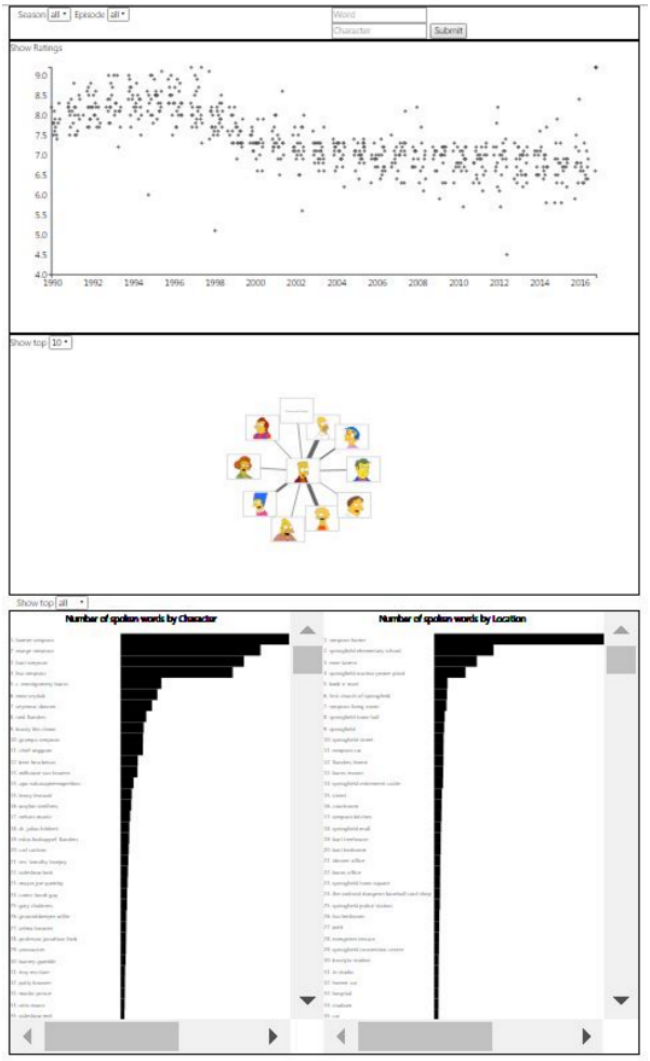


Figure 4: Hi - Fi prototype for the simsonyzer project.

3.2 Scatter plot

From the very beginning we decided to include a scatter plot into our visualization, because showing a correlation between air date and IMDB rating would cover two of four our main user tasks:

- Determine how various aspects of the series changed over the time
- Find out what are the best rated episodes

Moreover, a scatter plot was already a part of M3 and M4, and the evaluation of those showed mostly positive results except of one minor drawback like bad readability. Anyhow, we knew how to solve the readability problem.

Further we want to discuss why we decided to design the scatter plot the way we did it. As marks we decided to use simple circles for representing the episodes. We did not see any reason to go for another shape, like triangle or rectangle. One alternative would we dot, but dots are too small. Moreover, the users should be able to tell the circles apart from the background, that is why we

added thin boarders to the circles. One of the channels we used is the positioning of circles, both vertical and horizontal. Another channel was color for highlighting selected or about to be selected episodes/seasons. As according to T. Munzner combining position and hue results in best separability in terms of channel effectiveness. [2, p. 108]

Figure 5: Zoom function for the scatterplot.

T. Munzner states in her book that "Overviews are often shown at the beginning of the exploration process, to guide users in choosing where to drill down to inspect in more detail." [2, p. 136] Since this view is the first one the user sees, it was important for us to show an overview of the data and to encourage him or her for further exploration. That is why we decided to apply Ben Shneidermans design guideline of Overview First, Zoom and Filter, Details on Demand [3, pp. 336-343]. Since we have around 600 episodes to

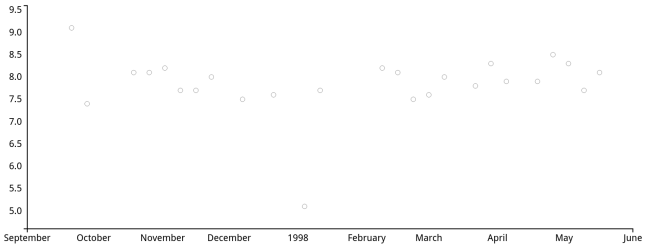


Figure 6: Zoomed in on one season.

plot, even though the selected episodes would be highlighted, it can become hard to read ratings for the user. Due to this reason if user clicks on the magnifying glass below each season, (fig. 5) the view will zoom in, displaying the zoomed-in season only (fig. 6). But in order to keep an overview and being able to compare selected episodes to other ones in context, user can zoom out by pressing a button that will appear. (fig. 7)

Figure 7: Zoom out button, to return to the overview.

3.3 Navigation and Filters

Because our users will be using the tool for fun, they will not want to invest time in learning how to use it. In order to keep things simple, we thought that less is more and decided to take the navigation widget from M2 dashboard 3 (fig. 8) for filtering by seasons and episodes. It was clear for us there was a drawback in this approach – users cannot select, for instance, multiple seasons/episodes. They can select only following:

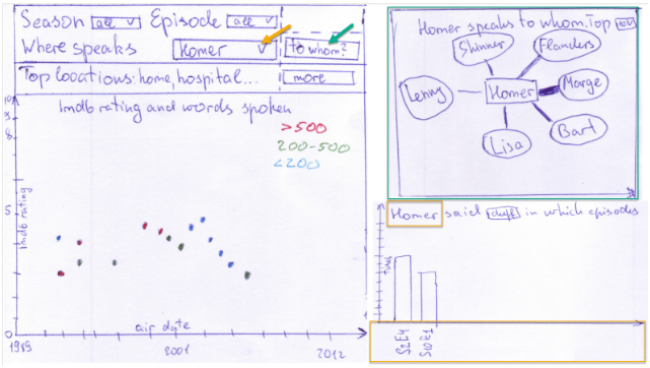


Figure 8: M2 dashboard 3 navigation.

- All seasons and episodes
- One season with all episodes in it
- One episode in one season

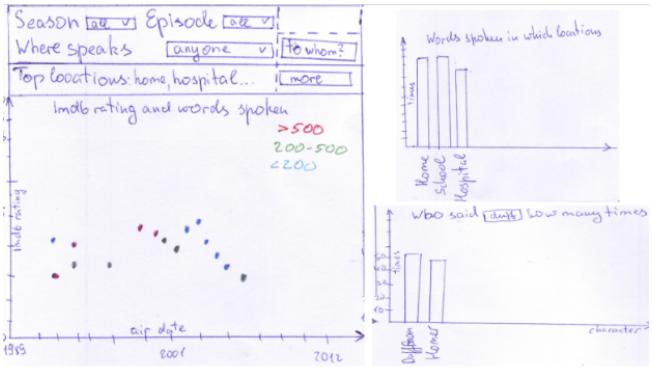


Figure 9: M2 dashboard 2

One of our ideas was to use the navigation view from dashboard 2 [M2](fig. 9), because we could have implemented selection of multiple random seasons/episodes by, for example, holding CTRL button. But since this feature is arguably unnecessary, because no meaningful information would be gained by just picking random episodes and plotting them onto the chart. That is why we thought this kind of selection would not interest our target audience and would overcomplicate usability. For a typical fan of The Simpsons the proposed method above would be sufficient enough.

Next we noticed that in every our dashboard from M2 filtering by spoken word and character is scattered all over each dashboard and thus is confusing for the users. It is hard to tell what will these filters apply to. That is why we decided to arrange filters according to the areas they are filtering. Moreover, if user scrolls down, the filters will be captured and anchored on the navigator bar. Respectively, if the the user scrolls up, the filters will be deattached. Last but not

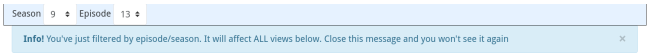


Figure 10: Info box under the filter.

least, after getting feedback on our final visualization tool which is described more detailed in 5.1, we understood it was not completely

clear for our test users that a filter is applied to all views below. Fortunately, we came up with a solution very quickly. After the user applies one of the filters an information box appears below the filter with a short explanation what the user just did. (fig. 10) The user can close it and it won't appear again.

3.4 Graph Who Speaks To Whom

Based on the fact that our visualization should be used for fun, we wanted to include at least one view that would comply with it. After doing some research we thought that a force directed graph would meet our needs, since it "[...] is relatively easy to understand and explain at a conceptual level, using the analogy of physical springs." [2, p. 204] In addition to that, a similar view to force directed graph was in every dashboard from M2 (fig. 3) and we included it into M3 as well. (fig. 11)

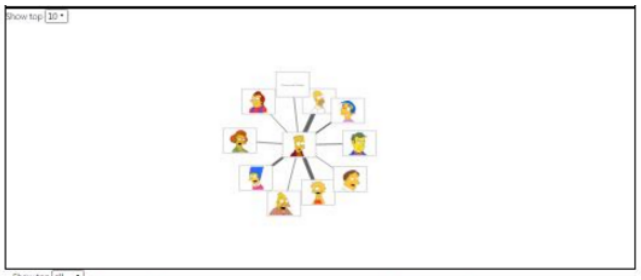


Figure 11: Graph in every meilstone and prototype.

As marks we used rectangles for the nodes and lines for the edges. Again, because this view should amuse the users and be pleasant to look at, pictures of some of the main characters were placed inside the rectangles. For the rest of the characters only the name will be shown. According to T. Munzner "Spatial position does not directly encode any attributes of either nodes or links; the placement algorithm uses it indirectly." [2, p. 204] That is why is was rather hard for us to use any channels. However, we used some of them, for example, for the width of the edges that correlates to the communication intensity between the two nodes. Moreover, we scaled the repelling force for the nodes, so that characters which communication intensity is stronger would attract each other.

If we compared this view to its prototypes from M2/M3, it underwent some changes. We included a dropdown element for showing not only top 10, but also top 15, 20 or 30. We decided to restrict the number of shown nodes in order to avoid the hairball effect. Additionally, we thought showing a high number of nodes would not interest the users. Also we decided to include a bar chart near the graph because of the feedback from test users [3]. The test users stated it was hard to tell what nodes were on what place, coming back to the major weakness of non-deterministic force-directed placement of the nodes. [2, p. 205] [4] In other words, sometimes they could not differentiate the width of the edges that denoted communication intensity between two characters/nodes. However, with a bar chart next to the graph, the presentation of the data would be both amusing to look at and accurate, as the users can very easily read from the barchart, for example, the number of the spoken words.

One of other challenges we needed to overcome was to properly fit the text inside the rectangles. (fig. 12) For M3 we tried to fit the text by scaling its font size, but in the end we decided to try to fit the

[3] More details in 5.1

[4] The final way of optimization we thought of was to draw every character floating around the selected one in ascending or descending order. Unfortunately, D3 did not allow us to determine the order of the nodes. D3 positions nodes always randomly and unpredictably, probably due to asynchronous nature of JavaScript.

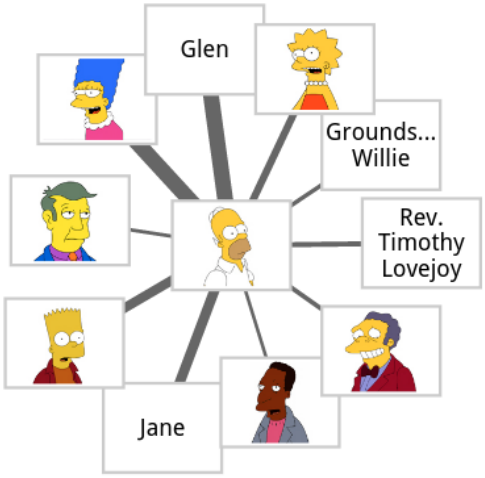


Figure 12: Name tags in the boxes.

text by using multiple lines which was a better idea after all. Our test users found it also more meaningful.

3.5 Barchart Spoken Word(s) by Character

In M2 we wanted to show how many words a character speaks in an episode or how many times a character says a certain word in an episode or both. Unlike the Graph Who Speaks to Whom, where one of our intentions was to have something our users will remember, this graph was not thought of to look fancy in any way. The fancy part of this graph was planned to come over the filters. We were looking for a simple and minimalistic way of showing what lies in the two-dimensional data. So we decided to use a barchart to fullfill this task, especially considering the data-ink ratio and trying to make it in a way where we "... prune out ink that fails to present fresh data-information." [4, p. 100]

Since the filters in our tool filter all views below, filtering works on this chart as following. The season and the episode filter are self-explaining and simply narrowing down the considered data and the word filter will only a certain word being counted. The character filter on the other hand changes the chart to not show any more the names of the different characters, but all episodes the filtered character appeared in and how many words or how many times a certain word has been spoken by this character in the selection.

In M3 we implemented this chart just as we imagined it to be: simple, self explaining and insight-offering in the data, even though a few things still were not optimal. Despite a big problem being the very slow performance, which will be discussed in 5.3, it was kind of hard to see what label belongs to which bar and where a bar ends and where the next one starts. Especially shorter names caused trouble of seeing what belongs where. Furthermore the user had to scroll down to see what the scaling of the current investigation is and even then it was most times impossible to tell how many words a character had said, since in most cases there was one dominating character who had very much more words than the characters at the bottom, so despite the user seeing the scaling at the bottom, the chance of telling how many words the characters at the bottom had spoken was not there. In our M3-feedback we were told even more problems of this chart, like the half words which appeared once the

user filtered in a way that showed results with a very low maximum of total words. For example if the Maximum was 1, the scaling would be shown as 0.0, 0.2, ... , 1.0.

In M4 we tried to solve all of these problems. Now there are gaps between every bar and the names of the characters and the episode numbers are now placed directly next, which makes it very easy to distinguish between the bars to the bars and very easy to determine which bar belongs to which label. The axis has been moved to the top of the chart, since the user wants to compare more likely the top results, especially after setting filters. To make the chart even more readable there are now tooltips showing the exact number of words, so it is now possible to tell how many words every character has spoken, no matter how little it was compared to the dominating character. Furthermore we made the header change every time a filter is set. The half words simply got fixed by scaling the axis with fewer ticks if the word-maximum is low. Another problem we described in the documentation of M3 was the fact, that combining all filters made this chart show only one bar and making it questionable redundant in this case. The test users we showed our tool to disagreed with us in this point. According to them "it is not unnecessary, because you still want to know how many words have been said by the character in this episode." For this reason we did not change what the chart is doing when many filters are combined.Then we also put more interaction into this chart by making it possible to click one episode or one character and then setting the filters directly like this.

Finally we reached a design for this chart where we have no ink, that is unnecessary in our opinion. The bars and the labels are undoubtedly important, even though the numbers next to the label may not be essential, they help determining the position of the character, even if the user decided to look at all results. This helps especially when there are no filter activated and the user wants to see at what place is a more uncommon character. The Axis is one thing that could have got rid of, like shown in [4, p. 128], but we decided to keep using the d3-method, that makes the axis itself, because we thought that non professional users might be more familiar with seeing axis. So if they see something that is not foreign to them, they might feel more comfortable with it. Also through giving the exact amount of words in the tooltips, there is no unnecessary ink, like it would be in alternative implementations, for example if the amount of words was always displayed at the top of each bar. (fig 3.6.1 and fig 3.6.2)

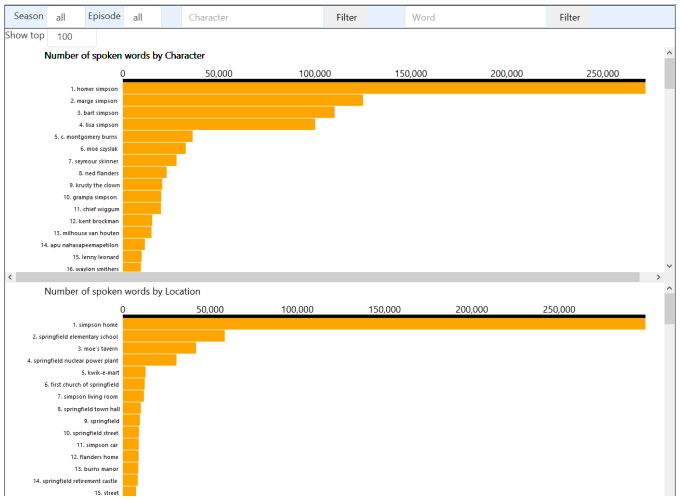


Figure 13: Two bar charts unfiltered.

3.6 Barchart Spoken Word(s) by Location

The intentions in M2 and problems in M3 in this chart were very similar to the intentions and problems of the Barchart Spoken Word(s) by Character. We wanted to give the user of the tool a way to explore how many words were said where. By applying filter the user can play around with the data and still enjoy this chart, even though it does not look special in any way.

Only the character filter does something else to this Barchart in comparison with the other 3 filter, that do exactly the same to this Chart like to the Barchart Spoken Word(s) by Character. Since here the data shown is based on the locations the character filter can here just narrow down the considered data, like the other 3 filter.

In M4 we solved the issues for this chart at the same time we solved the Problems of the other barchart. So there are no half words anymore, the readability has been enhanced by placing the labels better, putting a gap between all charts and by putting the axis to the top. Finally tooltips help getting the exact amount of words.

Since this chart is based on locations and there is no location filter, there are no filters that can be directly set in this chart, so we decided to not make it clickable. (13 and 14)

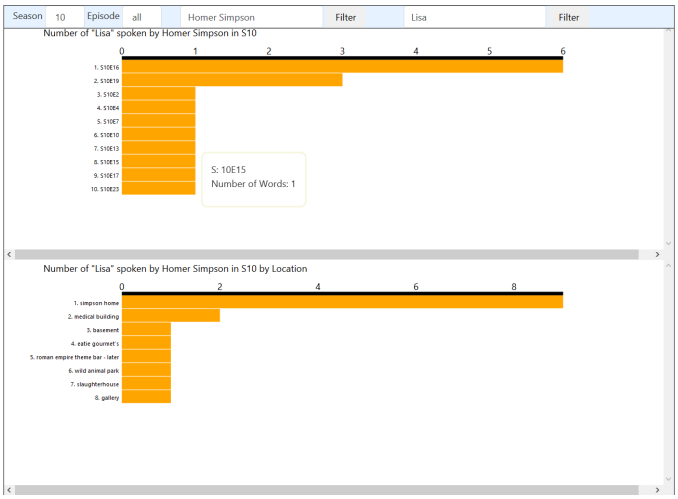


Figure 14: Two bar charts with some applied filters and a tooltip.

4 Implementation

The system was implemented mainly with D3.js visualization library written in JavaScript. The reason for this is because Tableau lacks some key features for our project. We would have to script them or find plugins that would do the job, but thanks to A3 we learned enough to feel confident about achieving our goals with D3.

Moreover, we used jQuery, because it makes manipulation of DOM elements significantly easier. To achieve better overall perception and ”pleasant look” we used Bootstrap 3 framework.

Additionally we developed a small application in Java for preprocessing the data for Who speaks to whom view. It can be found in the folder java.

One of the most serious implementation challenges we encountered was performance issue. We will discuss it more detailed in 5.3.

Unfortunately we figured out too late that the tool works properly only in Mozilla Firefox and we did not have time to solve this issue.

5 Results

5.1 Evaluation feedback

In order to evaluate our tool properly we collected feedback from four test users.[5] These users have no knowledge neither in computer science, nor in visualization techniques, but they are fans of the Simpsons, so they were the perfect test users for us. We did two test sessions with them: before starting implementing M4 and after the implementation was done. In every case every user had to accomplish our previously discussed main tasks 1.4. Further they were encouraged to try to get more from the tool. Our intention was to see whether our secondary tasks 1.4 will be discovered and fulfilled or not. Of course it was clear for us that the second session will not be as fruitful as the first one, because the user were familiar with it already.

During the evaluation process one of our team member was constantly noting everything the users did or said. As a result of the both sessions, none of the users had problems with following main tasks:

- Find out information about a specific episode or season
- Determine how various aspects of the series changed over the time
- Find out what are the best rated episodes

They could very quickly understand what to do and how to achieve the goals. However, one of the users had troubles with the task ”Find out interesting facts about the favorite character”. Our tool did not meet his expectations. He thought he will be presented with something like ”Have You Known” facts by selecting the character, rather than with what he got.

In terms of HCI during the first session three of the users stated it would be better to have the navigation elements always visible. One of them said it was not clear for him that a certain filter applies to all views below. Based on this feedback we did following changes for M4:

- The navigation bar got a fixed position above.
- The input fields got positioned right above the area they filter.
- The input fields got the ability to be anchored/deanchored to/from the navigation bar when user scrolls.
- User sees a one-time-message that describes what he or she just filtered and stresses out on the fact that this filter applies to ”ALL” views below.

[5] Age: 21-36, 3 males, 1 female

During the second session all users were very positively surprised by how much more convenient and transparent the navigation elements became. Even the one who did not complain before noticed big difference.

As far as the secondary tasks are concerned, the users had no major problems with further exploration of the tool’s functionality. Nevertheless, two of them stated it was hard read the graph Who Speaks To Whom. More specifically, they sometimes could not tell the variance of the edge density that denoted communication intensity. On the other hand, users liked dragging nodes around a lot. In order to solve this problem and keep letting users have fun with the dragging ability, we decided to place a barchart right next to the graph. During the second session users were satisfied with this problem’s resolution.

5.2 Scenarios of use

Even though we gathered feedback from real users, we still want to iterate on the two scenarios from M2/M3 in order to show standard use cases that should be relevant for any user from our target audience.

5.2.1 Scenario 1

Barney is a big Simpsons fan and is looking for episodes with his favorite character Barney in it.

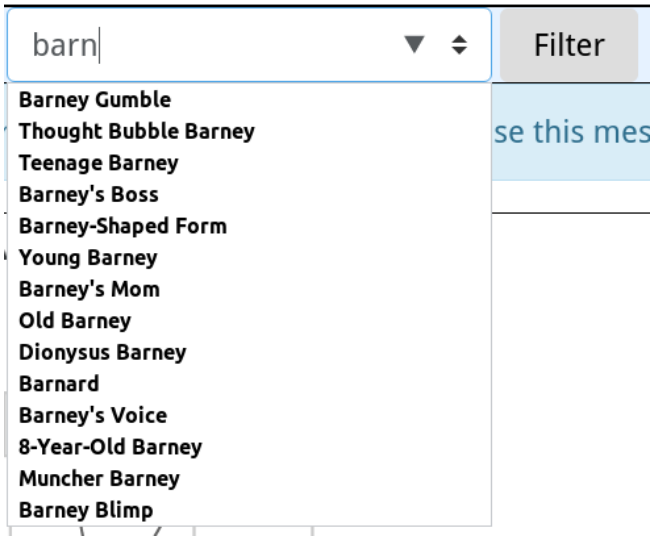


Figure 15: Character suggestion, with ajax.

To complete this task, Barney just has to type ”Barney” in the text field that says ”Character” and click ”filter”. (fig. 15) Instead of the funny placeholder (fig. 16) a graph will appears, showing who Barney speaks most often to. (fig. 17). The user can see in what episodes Barney speaks frequently by looking at the bar chart below.

Advantages:

- Only one action is necessary to fulfill this.
- Self Explaining. If you want information about a character and there is a text field, that says character, you will most likely try this first.

Disadvantages:

- It is not possible to search for two or more characters at the same time. So if Barney wants an episode where not only the character Barney, but also for example Skinner appears, he will have to do many searches and comparisons.

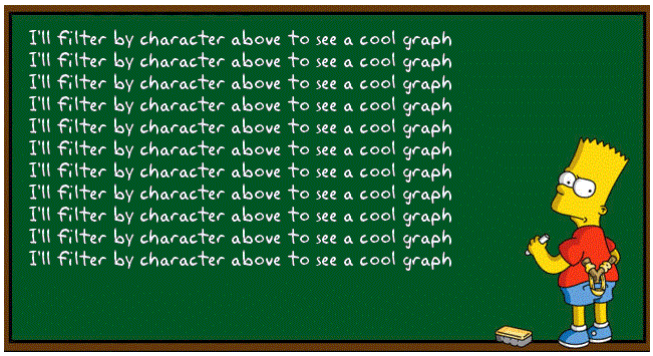


Figure 16: Reminder to select an character.

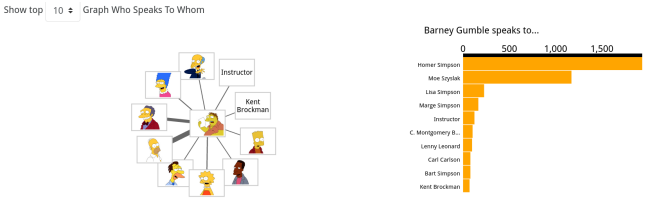


Figure 17: Barney selected, as character.

5.2.2 Scenario 2

Luke is a big fan of The Simpsons as well and hasnt watched them in years. He wants to know how the ratings have changed over the time. Moreover, he is a successful Silicon Valley businessman and doesnt have a lot of time, so he wants to know what are the best rated episodes and watch only them.

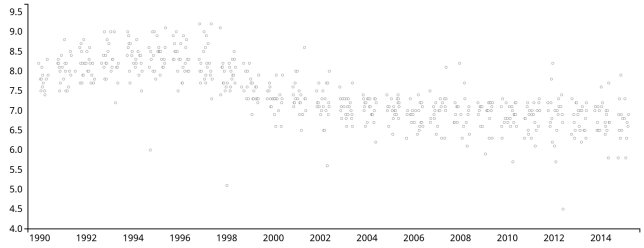


Figure 18: Scatterplot with IMDB ratings by time.

For this task not even a single click is necessary. All the information needed to fulfill this task is already displayed on the scatter plot (fig. 18) once the visualization tool is opened. All episodes are plotted according to their IMDB rating, so Luke can see all episodes with best ratings at one glance.

By hover over Luke can look up the name of an episode. If some episodes overlap each other and a particular episode becomes hard to be hover over it, Luke clicks on the magnifying glass below a highlighted season. The view is zoomed-in and now every episode is visible. (fig. 6)

Advantages:

- No interaction needed.
- User gets an overview.
- Zoom-in animation is pleasant to look at.

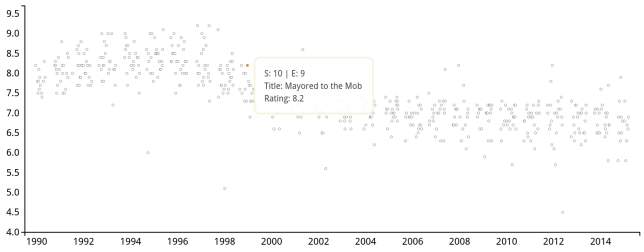


Figure 19: Scatterplot with tooltip.

Disadvantages:

- Only IMDB ratings are considered.
- Some of the dots overlap each other, however this problem is solved by implementing the possibility to zoom into a season.

5.3 Performance of the system

Overall we had very severe performance issues after implementing our Hi-Fi prototype. Not only initial tool booting, but also further views refreshing took almost 20 seconds to load. This was unacceptable and that is why we needed to find a solution. We had only two dynamic parts that could be the bottleneck: graph Who Speaks to Whom (fig. 17) and the two bar charts (fig. 13, 14) in the last rows. Since the algorithms used for plotting the graph had time complexity of O(log(n)) for the search and O(n) for the information collection, it was obviously not the problem. Our bottleneck was the many iterations over big arrays to find the entry where the word counts had to be incremented. Of course every character and location needs its own counter, but the implemented method of counting in M3 was not acceptable. No user would be able to enjoy the data when he/she has to wait up to 20-30 seconds to see the changes when applying a new filter. The idea we used to speed the implementation up was hashing. It was so obvious what to do, but still we did not think of it while developing M3. Every character has an id in the dataset and this id can simply be used as the key in the hashing process. Now since there are no iterations over big arrays user usually only waits a few seconds before the new charts show up, allowing them to stay focused on what they are trying to do and not discouraging them.

6 Discussion

Further we would like to present advantages and disadvantages of the Simpsonyzer. We aggregated them based on: the feedback we got from the test users; analysis of the standard use case scenarios; our own perception/wishes.

Advantages:

- Main tasks are easy to fulfill.
- The tool provides a good overview of the entire series.
- Self-explaining. If user wants to get information about a character and there is a text field with ”character”, he or she will most likely try it out.
- All filters can be combined.
- Info boxes with messages that appear depending on user’s actions serve as feedback and guidance.
- Smooth animations and highlighting makes users’ experience even nicer.

- The tool encourages user to try out something new he or she would not think about otherwise.

Disadvantages:

- Unfortunately we figured out too late that the tool works properly only in Mozilla Firefox and we did not have time to solve this issue.
- It is not possible to look for two characters at the same time.
- Using a graph for representing nodes that are connected to just one of them and are not interconnected with each other may not be the most optimal solution. Nevertheless, since our visualization tool supposes to amuse the users and our force directed graph not only looks fun, but can be dragged around, we decided this sacrifice is reasonable enough.
- More features could have been added and make the tool even better, but we still managed to implement everything we wanted.

7 Lessons learned

We had a chance to choose our own project topic, so we were very inspired to accomplish it properly. Alongside the things we heard during the course and while reading the papers given to us, which we also tried to use in our visualization tool, we learned a lot about the following.

Visualization: after finishing this project we can proudly say that we know the basics of Visualization and maybe even more. Based on first of all feedback from our test users we learned a lot of things by trial and error.

Graph: is was a real challenge for us to implement it. On the one hand we needed to avoid the hairball effect. On the other hand, we wanted to make the view fun, to please our target audience.

Performance: we had significant troubles in with it in M3 because. Our implementation was very slow. Not only we learned how to solve such a problem, we also figured out that is needed to think ahead regarding this problem, especially when it is necessary to iterate over a lot of data.

JavaScript/D3, because all of us were not very familiar with it and no one of us has ever used D3 before. When we worked with D3 for A3 we decided that we wanted to work with it instead of with Tableau, since then we have to learn scripting language of Tableau which is arguably not so popular. Now all three of us feel much more confident about D3 and with JavaScript working on a project. This is something that we already liked in HCI very much. We enjoyed the possibility to work with couple of iterations (M2/M3/M4). This way whole designing process is very well structured and brought closer to us students.

Here is an overview of who did what for the last milestone:

Who	What
Oleksandr Shabelnyk	• Graph 'Who Speaks to Whom', data preproccesing in Java, plotting the forced graph part • Interactions for Scatterplot (selection of episodes/seasons, magnifying glass animation, zoom-out functionality) • Navigation widget, dynamic anchoring when scrolling • Information boxes
Ivo Vidovic	• Barchart Spoken Word(s) by Character • Barchart Spoken Word(s) by Location • Barchart Who Speaks to Whom • Direct Filtering/tooltips in the Barcharts
Bernhards Bachinger	• Ajax-like search • Scatterplot, selection animation, zoom for the scatterplot

References

[1] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

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

[3] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996.

[4] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press LLC, 2001.