ArgumentNext – Visualizing arguments in the real world

Ana Sofia Silva
ana.sofia.m.s@tecnico.ulisboa.pt
Instituto Superior Técnico, Lisbon, Portugal

October, 2022

Abstract

Argumentation has been part of humanity's communication for centuries as a way of showing if someone is in favor or against an idea. However, analysing argumentative text is a complex task, due to its characteristics. To address this problem, it is necessary to search visual representations developed for this type of data as well as other visual techniques.

As part of the DARGMINTS project, the goal of this work is to create visualizations that facilitate the comprehension of argumentative text, both by analysts and non-experts. The solution consists in two platforms: the Dashboard and the ScrollyTeller. The Dashboard is a platform about opinion articles, with multiple interactive visual components and it was perceived as more confusing to non-experts comparatively to experts in the field, due to the concepts' complexity. In contrary, the ScrollyTeller, a storytelling visualization which uses information about parliamentary debates and visual components to tell a story, was well received by both, since the platform is more accessible.

Keywords: Argument visualization; Visual Analytics; Natural Language Processing; Text Visualization; Storytelling Visualization; Information Visualization.

1 Introduction

An argument is a statement or series of statements that is used to convince people about an idea or opinion. Argumentation has been present in humanity for centuries. Our society, specially politics, is built on argumentation: people try to convince others about their theories and others support or oppose to these by creating their own arguments.

To better understand these arguments, text is not enough. That's why that, in the nineteenth century, argument mapping was introduced [1]. However, the visual representation of arguments hasn't changed significantly since its inception.

Contrary to this, non-argumentative text visualizations have evolved a lot throughout the years. These tools help analysts find patterns as well as answer questions, for instance, about topics and stance evolution.

Having this in consideration, this project joins argument maps with what was learned from text visualizations throughout the years, thus creating new ways of visualizing arguments that are more intuitive and easier to understand.

This work is part of the DARGMINTS project that uses Natural Language Process algorithms and Argument Mining to extract arguments from texts and uses as case studies three different domains: opinion articles, parliamentary debates and political online forums.

The goal of this thesis is to **create a visualization for each case study with new and creative visual components that facilitate the analysis of argumentative text.** To accomplish this, a previously created framework, called Visual Argument (VA), is used as a foundation, which creates visual components for mock-up argumentative text.

Therefore, an adaptation to include real data produced by the DARGMINTS project as well as to contain new visual components is necessary. Since the case studies are different from each other, distinct demonstrators were developed, which show the versatility of the framework.

2 State of the Art

As noted in the previous section, it is important to look to argument visualization but to non-argumentative text visualizations as well. One type of text visualizations is storytelling visualizations.

Argument visualizations haven't changed substantially over the years and consist mainly in argument maps. An argument map is a diagram showing the logical structure of an argument. Typically it consists in nodes which represent propositions and links which correspond to relationships such as evidential support.

Argument visualization only emerged in the nineteenth century, and it was used by Richard Whately in 1836 in a logic textbook [1]. In the twentieth first century, different digital tools started to emerge.

Reason!Able [2] and Rationale [3], created by Tim van Gelder, in 2000 and 2006, respectively, and Convince Me, all use rectangles or ellipses to represent nodes and arrows or lines to show the various links, although having minor differences between them.

Even the most recent tools, like MindMup[1], use the same visual representations to show an argument.

Therefore, an argument map might be the best way to analyse the structure of an argument but it is not enough to understand if other characteristics might have influenced its creation. Non-argumentative text visualizations offer a panoply of representations to show text characteristics that might be present in argumentative text.

Tools for non-argumentative text explore different characteristics, such as stance, sentiment, topic and time in insightful ways.

For instance, ToPIN [4] visualizes time-anchored comments in online educational videos by focusing on the timecode of the comment, its topic, its type, and the sentiment behind it.

Other text visualizations explore text analysis in a deeper level, using linguistic measures such as the density of words in the text. Although the analysis itself does not concern this work, the used visual representations served as inspiration for the solution. Voyant Tools, for example, consists of 28 text analysis and visualization components[2]. Among them, there is the Collocates Graph, which consists in a forced network graph that demonstrates how terms in the text occur in close proximity.

Some of the text visualization present multiple visual components in a dashboard like layout, such as the MonkeyLearn[3] tool, which is a no-code interface that helps create visualizations using AI technology for text analysis.

Storytelling visualization were also explored since they are a creative way of telling a story based on visual components. Most of them use scrolling as a trigger event to change the state of the visualization.

According to Jeffrey Heer and Edward Segel [5], there are three types of storytelling visualizations. A Martini Glass visualization, initially, is guided by questions and observations created by the author, afterwards, the reader can interactively explore the shown visualization [4] [5]. An Interactive Slideshow, follows a slideshow structure, and provides some interaction mid-narrative, which allows the user to explore the data before moving on to the rest of the story [6] [7] [8]. Finally, a Drill-Down Story, presents a

[1]MindMup website: https://www.mindmup.com/

[2]List of Voyant Tools: https://voyant-tools.org/docs/#!/guide/tools

[3]MonkeyLearn: https://monkeylearn.com/

[4]Are Pop Lyrics Getting More Repetitive?: https://pudding.cool/2017/05/song-repetition/

[5]Can Data Die?: https://pudding.cool/2021/10/lenna/

[6]Twenty Years Of The NBA Redrafted: https://pudding.cool/2017/03/redraft/

[7]Homan Square: A portrait of Chicago's detainees: https://www.theguardian.com/us-news/ng-interactive/2015/oct/19/homan-square-chicago-police-detainees

[8]How China's economic slowdown could weigh on the rest of the world: https://www.theguardian.com/world/ng-interactive/2015/aug/26/china-economic-slowdown-world-imports

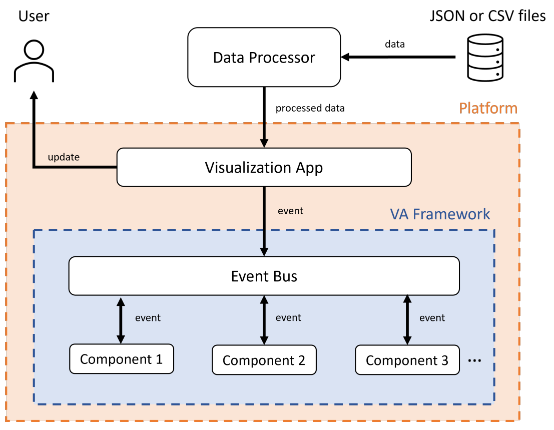


Figure 1: Architecture Diagram

general theme first and allows the reader to choose between instances of that theme to learn more about it [9]. This provides a way for the user to read the stories they want, when they want.

In conclusion, multiple visualizations of different types were explored, in order to make a creative solution. Argument maps remain an important visual component but time, stance and topic should also be analysed. Storytelling visualizations can also provide a better and more engaging way of telling a story taken from argumentative text.

3 Visual Argument Framework

Since there is not yet a platform that uses visual components to analyse all the characteristics present in argumentative text, there is a great potential in using the work seen in the state of the art as an inspiration to this framework.

This framework must provide visual components that interact with each other, whilst being independent and the platforms this framework creates must be usable by experts and non-experts in linguistics.

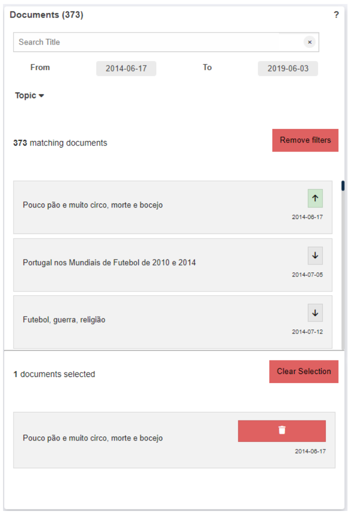
Since the first case study, opinion articles, refers to a deeper argumentative text analysis, which means the information regarding the structure of the argument will be more important, the platform needs to allow the analysis of the components of the argument and the evolution of the arguments throughout the documents.

Contrary to this, in the parliamentary debates case study, the goal is to tell a specific story using the provided data, so this demonstrator needs to be able to convey the story in the most precise and clear way possible, while providing means for scroll, zoom, hover and other forms of interaction.

3.1 Architecture

The developed demonstrators follow the architecture presented in Fig.1, where the VA Framework is shown as part of the web platform.

[9]OECD Better Life Index: https://www.oecdbetterlifeindex.org/


Documents (373) ?

Search Title

From 2014-06-17 To 2019-06-03

Topic

373 matching documents — Remove filters

Pouco pão e muito circo, morte e bocejo — 2014-06-17

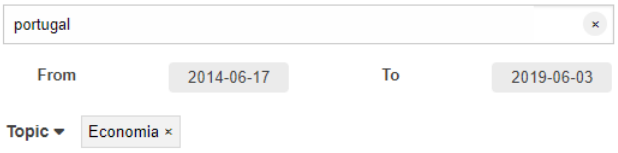
Portugal nos Mundiais de Futebol de 2010 e 2014 — 2014-07-05

Futebol, guerra, religião — 2014-07-12

1 documents selected — Clear Selection

Pouco pão e muito circo, morte e bocejo — 2014-06-17

Figure 2: Example of the list component


portugal

From 2014-06-17 To 2019-06-03

Topic Economia ×

Figure 3: Example of the filter bar component

The datasets are retrieved from JSON or CSV files and passed to the Data Processor, which is responsible for reading and processing the data, depending on the needs of the platform. The processed data is then passed to the Visualization App, which is specific to the platform, and it is in charge of creating the structure of the platform in question. After the platform structure is constructed, the Visualization App emits an event through the Event Bus, which, in turn, emits events to create the necessary components that exist in the VA Framework.

The Event Bus also permits the transmission of data between components which allows for great interactivity.

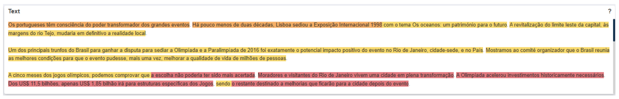
The VA framework is composed by various components, that will be listed next.

3.2 List Component

Since the provided dataset presents more than 300 documents, a list component, seen in Fig.2, was included, which is divided in three parts. The filters' bar will be explained next. The filtered documents list presents all the documents that match the filters. Finally, the selected documents list presents all the documents that were selected. To know how many documents are filtered or selected there are written indicators on top of both lists specifying it.

3.3 Filter Component

To easily search the various documents, there are a series of filters that can be used to filter the list of articles, as seen in Fig.3, such as the search bar used

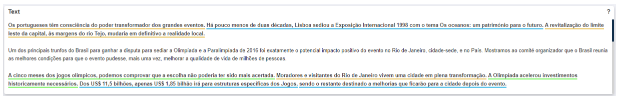

Text ?

Os portugueses têm consciência do poder transformador dos grandes eventos. Há pouco menos de duas décadas, Lisboa sediou a Exposição Internacional 1998 com o tema Os oceanos: um património para o futuro. A revitalização do limite leste da capital, às margens do rio Tejo, mudaria em definitivo a realidade local.

Um dos principais trunfos do Brasil para ganhar a disputa para sediar a Olimpíada e a Paralimpíada de 2016 foi exatamente o potencial impacto positivo do evento no Rio de Janeiro, cidade-sede, e no País. Mostramos ao comité organizador que o Brasil reunia as melhores condições para que o evento pudesse, mais uma vez, melhorar a qualidade de vida de milhões de pessoas.

A cinco meses dos jogos olímpicos, podemos comprovar que a escolha não poderia ter sido mais acertada. Moradores e visitantes do Rio de Janeiro vivem uma cidade em plena transformação. A Olimpíada acelerou investimentos historicamente necessários. Dos US$ 11,5 bilhões, apenas US$ 1,85 bilhão irá para estruturas específicas dos Jogos, sendo o restante destinado a melhorias que ficarão para a cidade depois do evento.

Figure 4: Example of the text component with all annotators selected


Text ?

Os portugueses têm consciência do poder transformador dos grandes eventos. Há pouco menos de duas décadas, Lisboa sediou a Exposição Internacional 1998 com o tema Os oceanos: um património para o futuro. A revitalização do limite leste da capital, às margens do rio Tejo, mudaria em definitivo a realidade local.

Um dos principais trunfos do Brasil para ganhar a disputa para sediar a Olimpíada e a Paralimpíada de 2016 foi exatamente o potencial impacto positivo do evento no Rio de Janeiro, cidade-sede, e no País. Mostramos ao comité organizador que o Brasil reunia as melhores condições para que o evento pudesse, mais uma vez, melhorar a qualidade de vida de milhões de pessoas.

A cinco meses dos jogos olímpicos, podemos comprovar que a escolha não poderia ter sido mais acertada. Moradores e visitantes do Rio de Janeiro vivem uma cidade em plena transformação. A Olimpíada acelerou investimentos historicamente necessários. Dos US$ 11,5 bilhões, apenas US$ 1,85 bilhão irá para estruturas específicas dos Jogos, sendo o restante destinado a melhorias que ficarão para a cidade depois do evento.

Figure 5: Example of the text component with one annotator selected

to search the title of the article. It is also possible to filter the list by date and topic. The filter component also interferes with the Network Graph Component.

3.4 Text Component

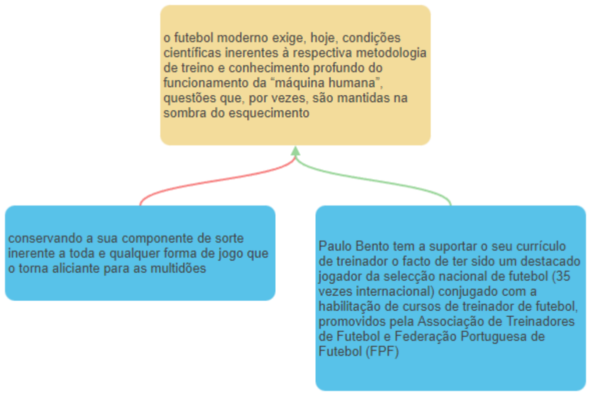
Since knowing the text of the selected documents is a crucial aspect of any tool of this type, as seen in the works in the state of the art section, a text component was included in the framework. This component shows the raw text of a document and other characteristics, such as the annotators. An annotator is the person who manually extracts the propositions that constitute an argument. This process is also called text segmentation and the text fragments are called Argumentative Discourse Units (ADUs).

Since each article has three annotators assigned to it and each created its list of annotations with argument maps, it was to be expected that some parts of the text had more common annotations than others. To indicate this, it was created a form of heatmap, where the most intense color shows the parts with more common annotations. Fig.4 presents an example with three paragraphs, where the last one has the most intense color, showing that it has more arguments in common.

The heatmap only appears when all the annotators are selected. If only one is selected, the component changes view, showing the annotations of the selected annotator underlined, seen in Fig.5.

3.5 Argument Map Component

As previously seen, the framework needs to allow the analysis of the argument structure, making it possible to identify the present ADUs and relations. As the related work indicates, this analysis is possible through the use of argument maps. In Fig.6, it is presented an example of an argument map developed by the VA Framework. The ADUs are represented as rectangles of different colors, which correspond to their types and the relations are represented as arrows of different colors, red for attack and green for support relations.


o futebol moderno exige, hoje, condições científicas inerentes à respectiva metodologia de treino e conhecimento profundo do funcionamento da "máquina humana", questões que, por vezes, são mantidas na sombra do esquecimento

conservando a sua componente de sorte inerente a toda e qualquer forma de jogo que o torna aliciante para as multidões

Paulo Bento tem a suportar o seu currículo de treinador o facto de ter sido um destacado jogador da selecção nacional de futebol (35 vezes internacional) conjugado com a habilitação de cursos de treinador de futebol, promovidos pela Associação de Treinadores de Futebol e Federação Portuguesa de Futebol (FPF)

Figure 6: Example of the argument map component

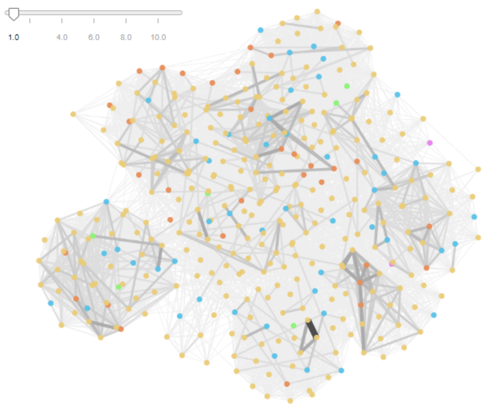


Figure 7: Example of the network graph component using Cytoscape.js

3.6 Network Graph Component

As seen in the state of art section, network graphs are frequently used to represent the connections between entities or keywords, depending on the data they are conveying. Since the provided data presents no information about the places and the people, the comparison of the documents can only be about the words that relate to the article or debate, depending on the dataset. Therefore, the connections will represent the similarity of the documents. Due to unsatisfying results in the first approach, two different libraries were used for each case study: D3.js[10] and Cytoscape.js[11]. In both representations, the documents correspond to nodes and the connections correspond to links. The stronger the link, the stronger the connection. In other words, if a link is darker and thicker, it means that the documents connected to that link are more similar.

In Fig.7, it is presented the network graph component using Cytoscape.js and in Fig.8, it is presented the network graph component using D3.js force model algorithm. Although they use different tools for the nodes layout, the behaviour is the same.

[10]D3.jshttps://d3js.org/
[11]Cytoscape.js: https://js.cytoscape.org/

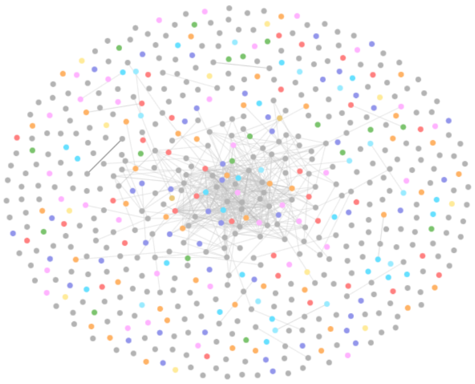


Figure 8: Example of the network graph component using D3.js

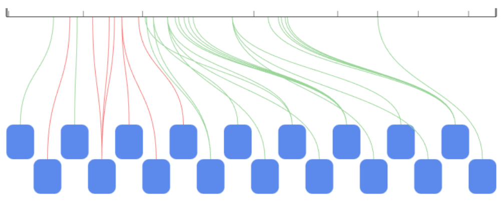


Figure 9: Example of the distribution component using the paragraphs in the axis

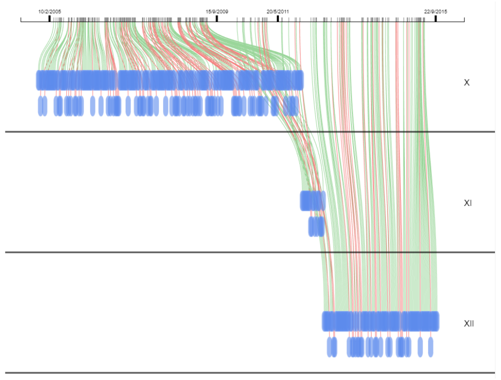


Figure 10: Example of the distribution component using the time in the axis

3.7 Distribution Component

One aspect that must be allowed by the framework is the ability of seeing the documents through time or, in a more atomic scale, the annotations throughout the document. Similarly to the ToPIN tool, seen in the state of the art section, this component consists in an axis, rectangles and links that connect the previous two.

In Fig.9, the component presents an axis that represents the length of the document. Each line separating the axis represents a paragraph. In Fig.10, the axis represents time.

3.8 MapOverview Component

Since the previous version of the VA Framework did not allow to analyse the structure of more than one argument map without scrolling up and down

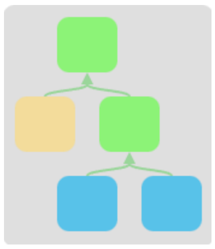


Figure 11: Example of the map overview component

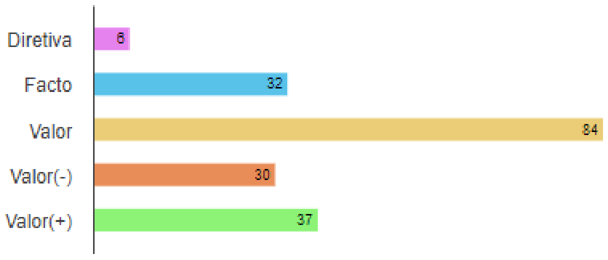


Figure 12: Example of the bar chart component

multiple times, a solution was created. This component consists in a small representation of the argument map, as presented in Fig.11, without its text, so, if wanted, the user can have a zoomed-in view of the argument map with the text.

3.9 BarChart Component

In the state of art, both text visualizations and storytelling visualizations resorted to simple charts to show fundamental information about the data they were conveying. It is not illogical to think that charts like bar charts, stacked bar charts, line charts, etc., can be useful to show data that are present in the provided datasets. Some of the requirements can be met with one of these straightforward charts. Although they are simple, the framework can improve it, by allowing interaction and customization. An example of this new component can be seen in Fig.12.

3.10 StackedBarChart Component

A stacked bar chart is another example of a simple chart, that can be used to convey useful information. In the example shown in Fig.13, the structure is similar to the Bar Chart Component, except for the two series, red and green, that constitute the bar. Each color has a meaning depending on the data that is being conveyed.

3.11 PositionChart Component

This component is a variation of the Stacked Bar Chart Component. While in the previous component it is already known by how many pieces the bar will be separated before hand, in this component it is dynamic, according to the selected documents.

The Position Chart Component was only used for the opinion articles. The component has two bars,

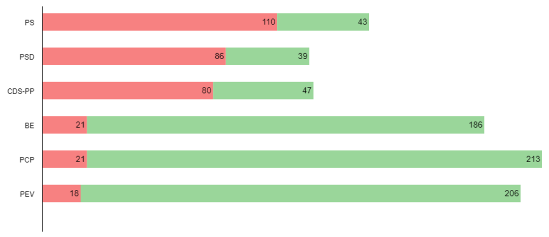
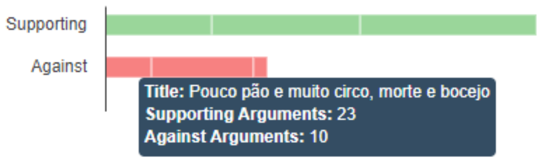


Figure 13: Example of the stacked bar chart component



Title: Pouco pão e muito circo, morte e bocejo
Supporting Arguments: 23
Against Arguments: 10

Figure 14: Example of the position chart component

separated by white lines; each piece represents a selected document. In Fig.14, the example shows the component with three documents selected.

3.12 LineChart Component

The Line Chart Component is fundamentally a simple line chart that allows to see the evolution of anything through time.

In Fig.15, there are two axis, the vertical axis, with the percentage values, and the horizontal axis, with time values – in this case, referring to legislatures.

4 Visual Argument Platforms

To show the versatility of the Visual Argument Framework, two distinct demonstrators were created for the two case studies. The first platform to be developed was the Dashboard, which uses data about opinion articles, and focuses more in the analysis of arguments. The second platform is the ScrollyTeller,

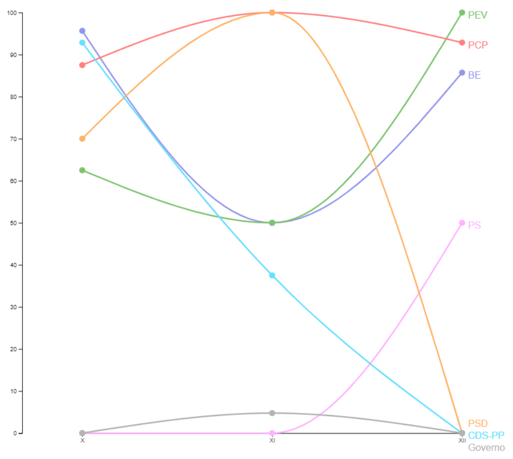


Figure 15: Example of the line chart component

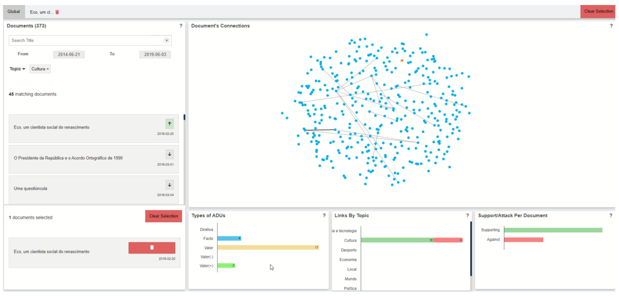


Figure 16: Screenshot of the global view of the Dashboard

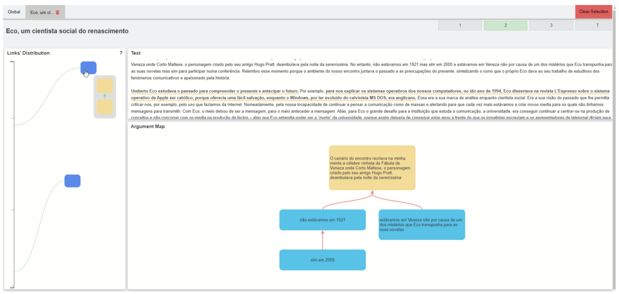


Figure 17: Screenshot of the specific view of the Dashboard

a storytelling visualization about parliamentary debates, with the goal of being more accessible to non-experts on the linguistics field.

4.1 Dashboard

The structure of the Dashboard consists in two views: a global and a specific view, shown as tabs, both constituted by compartments that lodge the visual components created by the VA Framework.

The global view, as seen in Fig.16, presents the list component on the left, showing the multiple documents on the dataset, including optional filters.

On the right side, in the top row, there is the network graph component showing the connections between these documents according to their keywords. However, as seen in the picture, this component was not giving good results, as it was not possible to take significant conclusions about the connections of the documents. A solution for this problem will be discussed next.

Concluding this view, in the bottom row, three similar charts are presented: the bar chart, the stacked bar chart and the position chart. The bar chart presents the distribution of the types of ADUs in the selected documents. The stacked bar chart shows how many support and attack relations there are for each topic in the selected documents. Finally, the position chart shows how many support and attack relations each selected document has.

The specific view appears when a document is selected in the list and it only refers to that specific document. As seen in Fig.17, this view consists in three components with a group of buttons on the top-right side of the screen.

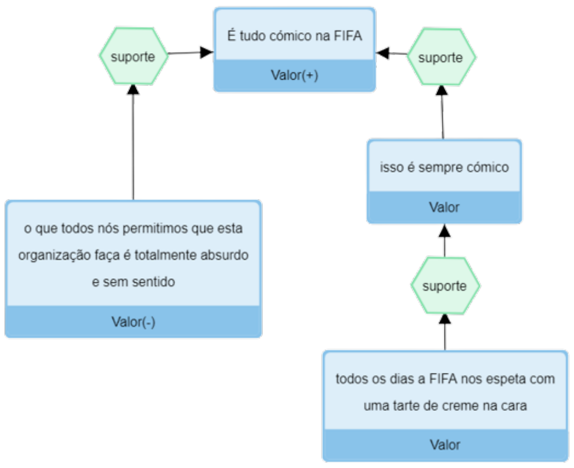

suporte
É tudo cómico na FIFA
Valor(+)
suporte
isso é sempre cómico
Valor
o que todos nós permitimos que esta organização faça é totalmente absurdo e sem sentido
Valor(-)
suporte
todos os dias a FIFA nos espeta com uma tarte de creme na cara
Valor

Figure 18: Screenshot of argument map in dataset format

The group of buttons refers to the annotators of the selected article. The buttons 1 to 3 correspond to the annotators 1 to 3, while the button "T" corresponds to all the annotators – "Total". By clicking on one of these buttons, it is possible to see the annotations of a specific annotator or of all of them. This affects both this view and the global one.

On the left side of this view, the distribution component shows the distribution of relations throughout the document. The axis is separated in lines that represent the end of a paragraph. Each rectangle represents an argument map and by clicking on it, the corresponding argument map is shown. The connections between the rectangles and the timeline correspond to the relations present in that map, each with the color associated with their type.

The full text of the document can also be analysed. This component, as previously explained, changes according to the option selected in the group of buttons.

4.1.1 Problems in the implementation

During the development of the Dashboard, there were a few setbacks. For instance, regarding the integration of the opinion articles dataset, there was a complication concerning the argument maps, since these were in a different structure in the dataset than the framework was expecting.

In Fig.18 it is shown an example of an argument map in the structure provided by the dataset. The different map layouts caused some conflict in the framework, since it was not prepared for this form of data. As it is possible to see in Fig.18, the argument maps in the dataset were composed by extra nodes that indicated the nature of the link between two nodes. This varies from the expected layout, where the arguments maps are constituted by nodes with the information necessary to know if they are of support or attack, and where they connect directly to each other without extra nodes in the middle.

Therefore, the framework, mainly, this component, had to be adapted, which required some complex changes to translate from one layout to other. In particular, it was necessary to pass the information from the extra nodes to the information nodes, to have the desired layout.

The remaining components did not have problems with data integration, except for the map overview, which is similar to the argument map component.

Another problem that occurred during the platform's implementation concerns the network graph that, when using data about opinion articles, was not giving significant results. The D3.js force model algorithm did not seem to be applying the force necessary to approximate the most similar nodes as can be seen by the long strong lines present in the graph, as seen in Fig.16. Even after altering some parameters in this tool's configuration, the structure of the network graph did not change.

Thus, other solutions were explored regarding the construction of network graphs. Cytoscape.js was one of the tools found that could improve the structure of the graph, while being easily integrated with the already existing libraries in the project. Cytoscape.js[12] is a JavaScript library used for graph analysis and visualization that allows users to easily display and manipulate rich and interactive network graphs. This library granted the same interaction level as D3.js.

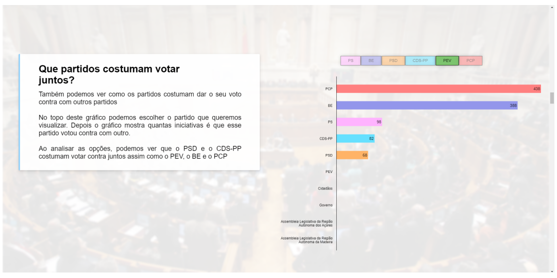
After integrating this library within the framework, the graph using data from opinion articles had more important results. As seen in Fig.7, there are notable clusters representing articles that depict similar subjects. This is due to the force applied on the nodes – when the link is stronger, the nodes are closer to each other.

The use of this library, however, has a downside: the loading time, which is significantly longer than when using D3.js.

4.2 ScrollyTeller

The ScrollyTeller was the second demonstrator to be created, and uses data about parliamentary debates from 2005 to 2015, which corresponds to legislatures X, XI and XII. While comparing the three legislatures, which faced entirely different circumstances regarding their govern, it was possible to analyse patterns and the evolution of the parties' initiatives. After some examination, using the Flourish[13] tool to generate instantaneous charts to easily observe these patterns, evidence that initiatives got more rejected in the first legislature, where Partido Socialista had absolute majority, became clearer. Therefore, the premise of the story turned into if absolute majorities indicated more rejected initiatives or not.

Since the ScrollyTeller is a storytelling visualization, it is expected to combine text and visual component to tell this story. This visualization focuses


Que partidos costumam votar juntos?

Também podemos ver como os partidos costumam dar o seu voto contra com outros partidos

No topo deste gráfico podemos escolher o partido que queremos visualizar. Depois o gráfico mostra quantas iniciativas é que esse partido votou contra com outro.

Ao analisar as opções, podemos ver que o PSD e o CDS-PP costumam votar contra juntos assim como o PEV, o BE e o PCP

PS
BE
PSD
CDS-PP
PEV
PCP
PCP
438
BE
388
PS
98
CDS-PP
82
PSD
68
PEV
Cidadãos
Governo
Assembleia Legislativa da Região Autónoma dos Açores
Assembleia Legislativa da Região Autónoma da Madeira

Figure 19: Screenshot of the bar chart component with buttons, with option "PEV"

on a specific story while allowing the users some interactivity mid-narrative, so it is possible to explore the data before moving on to the rest of the story. Relating this to the three approaches seen in the state of the art section, the platform corresponds to a Interactive Slideshow.

Furthermore, for the more curious users, an area is provided to allow the exploration of data that is not at all present in the story.

To allow this structure, a library with the same name was used[14], which is a JavaScript library that creates the HTML elements that allow the scrolling of data, which is retrieved from CSV files.

The library also allows to easily change the graphs of each story section while the user is scrolling and the story is being told. Therefore, the usage of the ScrollyTeller library allowed a development more focused on the visual components of the VA Framework and their integration with another type of visualization than with HTML elements and scrolling events.

Having in mind the chosen structure, library and story, it was possible to develop a prototype for the ScrollyTeller platform. This demonstrator starts with the initial page, stating the title of the story. Afterwards, it is presented a brief introduction about the data used for the story, in order to contextualize the user about it.

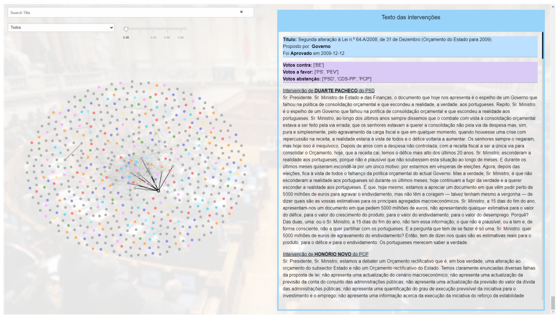
Following this, the platform maintains the same pattern: container of text on the left accompanied with a graph on the right.

Most of the visual components that were used in the Dashboard demonstrator were used in the ScrollyTeller, such as the bar chart and the stacked bar chart, which can be seen in Fig.19. The distribution component was also used but it was adapted, since it was conveying more data. For this platform, it was created a line chart to convey the necessary data.

Some components changed through scroll, as allowed by the library, and others changed through buttons, which is the case of the presented bar chart, in Fig.19.

As previously described, this platform includes an exploratory area, similar to the Dashboard demonstrator, which consists mainly in a network graph and a text box. This area can be seen in Fig.20,

[12]Cytoscape.js: https://js.cytoscape.org/
[13]Flourish: https://app.flourish.studio/
[14]ScrollyTeller: https://github.com/ihmeuw/ScrollyTeller



Texto das intervenções

Titulo: Segunda alteração à Lei n.º 64-A/2008, de 31 de Dezembro (Orçamento do Estado para 2009).
Proposto por: **Governo**
Foi **Aprovado** em 2009-12-12

Votos contra: ['BE']
Votos a favor: ['PS', 'PEV']
Votos abstenção: ['PSD', 'CDS-PP', 'PCP']

Intervenção de **DUARTE PACHECO** do PSD
Sr. Presidente. Sr. Ministro de Estado e das Finanças, o documento que hoje nos apresenta é o espelho de um Governo que falhou na política de consolidação orçamental e que escondeu a realidade, a verdade, aos portugueses. Repito, Sr. Ministro: é o espelho de um Governo que falhou na política de consolidação orçamental e que escondeu a realidade aos portugueses. Sr. Ministro, ao longo dos últimos anos sempre dissemos que o combate com vista à consolidação orçamental estava a ser feito pela via errada; que os senhores estavam a querer a consolidação não pela via da despesa mas, sim, pura e simplesmente, pelo agravamento da carga fiscal e que em qualquer momento, quando houvesse uma crise com repercussão na receita, a realidade estaria à vista de todos e o défice voltaria a aumentar. Os senhores sempre o negaram, mas hoje isso é inequívoco. Depois de anos com a despesa não controlada, com a receita fiscal a ser a única via para consolidar o Orçamento, hoje, que a receita cai, temos o défice mais alto dos últimos 20 anos. Sr. Ministro, esconderam a realidade aos portugueses, porque não é plausível que não soubessem esta situação ao longo de meses. E durante os últimos meses quiseram escondê-la por um único motivo: por estarmos em vésperas de eleições. Agora, depois das eleições, fica à vista de todos o falhanço da política orçamental do actual Governo. Mas a verdade, Sr. Ministro, é que não esconderam a realidade aos portugueses só durante os últimos meses; hoje continuam a fugir da verdade e a querer esconder a realidade aos portugueses. É que, hoje mesmo, estamos a apreciar um documento em que vêm pedir perto de 5000 milhões de euros para agravar o endividamento, mas não têm a coragem — talvez tenham mesmo a vergonha — de dizer quais são as vossas estimativas para os principais agregados macroeconómicos. Sr. Ministro, a 15 dias do fim do ano, apresentam-nos um documento em que pedem 5000 milhões de euros, não apresentando qualquer estimativa para o valor do défice, para o valor do crescimento do produto, para o valor do endividamento, para o valor do desemprego. Porquê? Das duas, uma: ou o Sr. Ministro, a 15 dias do fim do ano, não tem essa informação, o que não é plausível, ou a tem e, de forma consciente, não a quer partilhar com os portugueses. E a pergunta que tem de se fazer é só uma, Sr. Ministro: quer 5000 milhões de euros de agravamento do endividamento? Então, tem de dizer-nos quais são as estimativas reais para o produto, para o défice e para o endividamento. Os portugueses merecem saber a verdade.

Intervenção de **HONÓRIO NOVO** do PCP
Sr. Presidente, Sr. Ministro, estamos a debater um Orçamento rectificativo que é, em boa verdade, uma alteração ao orçamento do subsector Estado e não um Orçamento rectificativo do Estado. Temos claramente enunciadas diversas falhas da proposta de lei: não apresenta uma actualização do cenário macroeconómico; não apresenta uma actualização da previsão da conta do conjunto das administrações públicas; não apresenta uma actualização da previsão do valor da dívida das administrações públicas; não apresenta uma quantificação do grau de execução previsível da iniciativa para o investimento e o emprego; não apresenta uma informação acerca da execução da iniciativa do reforço da estabilidade

Figure 20: Screenshot of the exploratory area of the ScrollyTeller

which presents, on top, a series of filters that can be applied to the network graph below. This graph presents all the initiatives, in a force model algorithm created by D3.js. Contrary to what happen in the Dashboard, this force model algorithm had good results with this dataset and it was not necessary to use the Cytoscape.js library.

When a node is clicked on the graph, it is highlighted, and the details regarding that initiative, such as the voting results and the debate interventions, appear on the right side. This area allows the user to know all the information of all the initiatives present in the dataset.

4.2.1 Problems in the Implementation

The construction of this demonstrator was significantly easier and faster than the first. This relates to the fact that by this time the VA Framework was already improved, due to the development of the Dashboard. Because of the framework's versatility, it was easily adapted to accommodate another dataset, in order to create the ScrollyTeller.

Nevertheless, due to the fact the data was divided in multiple files, it was harder to pass it to the platform in the format they were in. To address this problem, the files were processed, using Python[15] scripts. These scripts were able to create new datasets with the needed fields from two different datasets or with completely new computed fields that were later used by the platform. This processing was done previous to the development of the platform to save time in its loading.

The difficulties regarding the development of this platform itself concerned external aspects, such as the used library. Every time an unknown library is used, it is necessary a certain time to learn how to work with it. Specific features of this tool were more complex to understand, which slowed the development time.

Another struggle when creating the ScrollyTeller was writing the story. Since it portrays a complex subject, the story needed to be written with care and preciseness, which required time and political knowledge.

[15]Pyhtonhttps://www.python.org/

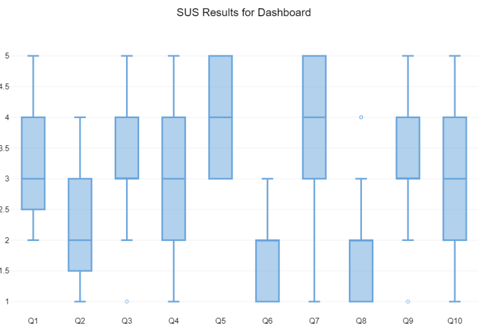


Figure 21: Dashboard's SUS survey results in the first testing phase

5 Evaluation

In order to confirm if the requirements were met and if there were improvements to be made, it was important to evaluate the developed platforms. Therefore, the testing plan was divided in two phases. The first phase corresponded to a usability testing with non-experts in the linguistics field. In the second phase, the tests were conducted with people from the DARGMINTS project, which are experts in the platform's domain.

5.1 Tests with Non-Experts

The first phase counted with twenty users, dispersed through various age groups.

Regarding the Dashboard, the users revealed more difficulties when using the network graph component, since they usually were not sure what it was conveying, specially, what the connections were representative of.

These users also struggled with the concept "annotators", consequently, selecting an annotator was not intuitive since they could not find the respective buttons. Related to this, the heatmap present in the text component was difficult to understand, since there was no indication about what the colors were conveying.

Overall, users felt there was a lack of captions and text boxes explaining the components to help them understand what each component was conveying.

As can be observed in Fig.21, the results of the SUS survey present some divergence in the users' opinions. Users had various struggles when performing the tasks, which made them feel they were not prepared for such tool. Nonetheless, most users did not think the platform was unnecessarily complex. The SUS score for the Dashboard testing was 62.9, which is below average and corresponds to a C-, on a scale from A+ to F.

Regarding the ScrollyTeller, many users faced struggles due to minimal orthographic errors, lack of punctuation or lack of precise terms in the story. Concerning the visual components, the users felt there was a need for a tooltip in the line chart to see the percentages of all points. In the exploratory

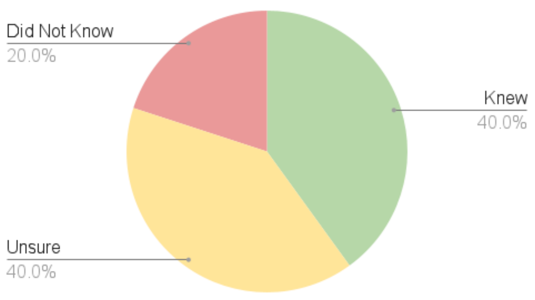


Figure 22: ScrollyTeller Task 8 answers

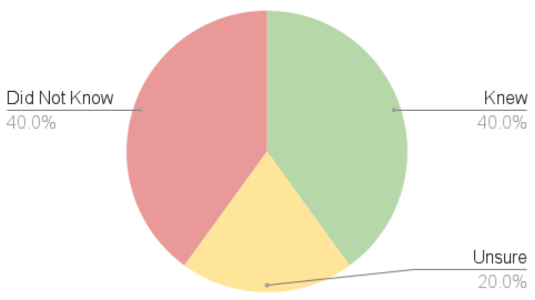


Figure 23: ScrollyTeller Task 6 answers

area, they struggled to notice the graph updated after searching the title, since the node was not highlighted enough.

The purpose of the last task was to see if the users understood the story, mainly its conclusion. As can be seen in Fig.22, when asked if there was a higher percentage in rejected initiatives in legislature X than in the remaining legislatures, 40% of the users said yes, while other 40% said yes, but was unsure about it and had to check in the platform. The rest of the users did not know the answer.

Regarding the SUS survey, the users felt that the platform was easy to use and it was not unnecessarily complex, which resulted in a score of 83.9, which is above average, and corresponds to an A.

5.2 Tests with Experts

The second phase of testing focused on the DARGMINTS project's requirements and if the features of the platforms were fulfilling them. This phase counted with five experts and used the method "think aloud", where the users say what they are thinking and doing while performing the tasks.

Although the experts had less difficulties, they explained they understood how a non-expert could be lost in this platform due to the complexity of the concepts related to argumentation. Due to this, they suggested the platform could have a form of tutorial in the beginning explaining what the components were conveying.

For these users, the platform felt easy to use and its functionalities seemed well integrated. However, they encountered the same problems seen in the first testing phase. This resulted in a SUS score of 66, which corresponds to a C.

Regarding the ScrollyTeller, the difficulties they faced were also seen in the first stage of testing, with non-experts, such as the ones related to inconsistencies in the writing of the story.

Another aspect these users discussed was the fact of getting frustrated by having to scroll up and down to see a certain part of the story. To address this, they suggested having a form of index that could accompany them through the story and allow them to jump from part to part. Consequently, an index could make ScrollyTeller be seen as a consultation platform instead of a website article that they only visit once.

Concerning the last task, as seen in Fig.23, 40% of the users knew that there was a higher percentage of rejected initiatives in legislature X than the remaining ones, while 20% were not sure about and had to check this information in the platform. The rest of the users did not know the answer.

The users agreed the platform was easy to use and its functionalities were well integrated, without many inconsistencies, besides the writing ones. They stated this platform was more accessible to other people, since it did not address complex concepts related to argumentation. The SUS result for the ScrollyTeller in this phase was 88.5, which corresponds to A+, which is the highest grade.

5.3 Discussion

Comparing the first and second phase of testing, it is possible to observe the platforms had similar results, according to the difficulties the users faced when navigating through them.

Regarding the Dashboard, the biggest difficulties were related to the lack of captions that would aid in associating the colors, sliders and highlights to the entities they are conveying. This platform already portrays complex concepts about argumentation, so captions are important.

Contrary to this, the ScrollyTeller, was seen as a more accessible and easy platform, both by non-experts and experts, since it does not require previous knowledge of complex concepts about argumentation. Although many users complained about the writing of the story, the majority of the users, in both testing phases, understood the story, and its conclusion.

This evaluation was able to prove how versatile the VA Framework is, since it was possible to create two completely distinct platforms that had different impacts and opinions from users.

5.4 Improvements

To answer to the comments received on both testing phases concerning the Dashboard, captions were created in the platform indicating the information that certain aspects, such as the color, were conveying. This would help in certain parts of the Dashboard such as the buttons for the annotators, the colors for the ADUs and relations and the highlights for the heatmap. Pop-up text boxes were created to explain each component and the slider was also changed, according to the received feedback, which made it more intuitive. This demonstrator also received multiple aesthetic changes, to make the platform more pleasing. The final version of the Dashboard can be found in the following link: https://web.ist.utl.pt/ist189407/DARGMINTS/

Concerning the ScrollyTeller, the writing of the story was the most commented aspect. To improve this, the story was rewritten and the text was edited with bold and colored excerpts to highlighted important statements or titles. Regarding the visual components, a tooltip was added to the line chart component that allowed to see all the values shown, by hovering over each point. About the exploratory area, when a title is searched, a list of suggested titles, compatible with the searched one, appears. When the user searches one title, the filtered nodes appear with a black border so they are more noticeable than before. The final version of the ScrollyTeller can be found in the following link: https://scrollyteller.herokuapp.com/.

6 Conclusion

Despite the fact argumentation is an important aspect in our daily lives, analysing argumentative text is not a simple task, due to the complexity and density of arguments. This thesis, aligned with the DARGMINTS project, presented the objective of creating a solution that would allow both experts and non-experts from the linguistics field to easily understand and explore this type of text.

To achieve this, two web platforms would be created, using different structures, demonstrating the versatility of a framework used for argumentative text. To create the two demonstrators, a search for visual representations to show the structure of arguments started. The most common are argument maps, which are essentially concept maps, but the nodes are connected by logical links. Since the visualizations surrounding argumentation are scarce, research turned to text visualizations, which are more insightful. A type of text visualization is called storytelling visualization, which tells a story using visual components.

After noticing some crucial elements that needed to be included in the VA Framework, it was possible to establish the requirements for both demonstrators. From there, the framework started to be altered and the demonstrators started to be developed, using various approaches. During the development process, the DARGMINTS project team provided feedback about the platforms during weekly meetings, which helped guide the solution to its best outcome.

In order to evaluate the web platforms, there was a two-phased testing, which consisted in testing the demonstrators with experts and non-experts in the linguists field. Overall, the Dashboard was perceived has a more complex platform than the ScrollyTeller because of the usage of argumentation concepts in the first demonstrator.

The Dashboard was altered mainly to be more intuitive, especially to non-experts, whilst the ScrollyTeller received changes related to the writing style to make the story more understandable and precise.

With the information collected from the tests, it was possible to pinpoint some interactions or features that the users would like to see in both platforms and could be assigned as future work.

Concerning the Dashboard, the users would like to be able to rearrange the various components to their liking. It would be interesting to create this feature in this platform to make it more personalized. Other interactions were missed by the users, such as a way to select an ADU and show it in the text component. Furthermore, a new mode of selecting documents could be explored, so it is not necessary to go to the opened tab after selecting a document. Having a way of selecting a document while staying in the global tab would be beneficial.

Regarding the ScrollyTeller, as suggested by the experts, an index would be a great addiction to this platform, since it would allow the user to go back and forth in the story without having to scroll to the parts that interest them.

References

[1] Pashler, H. *Encyclopedia of the Mind*. SAGE, 2013, pp. 50–52.

[2] Gelder, T. van. “Argument Mapping with Reason!Able”. In: *The American Philosophical Association Newsletter on Philosophy and Computers* 2 (2002).

[3] Gelder, T. van. “The rationale for Rationale™”. In: *Law, Probability and Risk* 6.1-4 (2007), pp. 23–42. ISSN: 1470-8396. DOI: 10.1093/lpr/mgm032. URL: https://doi.org/10.1093/lpr/mgm032.

[4] Sung, C.-Y., Huang, X.-Y., Shen, Y., Cherng, F.-Y., Lin, W.-C., and Wang, H.-C. “ToPIN: A Visual Analysis Tool for Time-Anchored Comments in Online Educational Videos”. In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’16. San Jose, California, USA: Association for Computing Machinery, 2016, 2185–2191. ISBN: 9781450340823. DOI: 10.1145/2851581.2892327. URL: https://doi.org/10.1145/2851581.2892327.

[5] Segel, E. and Heer, J. “Narrative visualization: Telling stories with data”. In: *IEEE transactions on visualization and computer graphics* 16.6 (2010), pp. 1139–1148.