MASTER EN SCIENCES INFORMATIQUES

STAVIZ

Application of Text Visualization Techniques to Structural Analysis

Clarinval, Antoine

Award date:
2017

Awarding institution:
Universite de Namur

Université de Namur
Faculty of Computer Science
Academic Year 2016–2017

STAVIZ : Application of Text Visualization Techniques to Structural Analysis

Antoine CLARINVAL

Internship mentor: Pr. Anne Wallemacq

Supervisor: ____________________________ (Signed for Release Approval - Study Rules art. 40)
Pr. Bruno Dumas

Co-supervisor: Pr. Anne Wallemacq

A thesis submitted in the partial fulfillment of the requirements
for the degree of Master of Computer Science at the Université of Namur

Abstract

Over the past decades, there has been a growing interest for text visualization techniques. Their goal is to provide text analysts with representations from which they can get knowledge that is tedious to get with the raw text only. This work aims at understanding how text visualization techniques can deliver these services to structural analysts. In doing so, this work presents STAVIZ, a software designed to help structural analysts in their work. Subsequently, STAVIZ is evaluated according to a proposed methodology and lines of thoughts for improvements are given. Ultimately, the hope is to lay the groundwork for future work and to provide useful guidelines to anyone willing to undertake the development of tools for structural analysts.

Ces dernières décennies, on a pu observer un intérêt grandissant pour les techniques de visualisation de texte. Leur but est de fournir aux analystes des représentations à partir desquelles ils pourront dégager des connaissances fastidieuses à obtenir à partir du texte brut seul. L'objectif du présent travail est de comprendre comment les techniques de visualisation de texte peuvent rendre ces services aux analystes pratiquant l'analyse structurale. Ce faisant, il présente STAVIZ, un logiciel créé dans le but d'aider ces analystes dans leur travail. Ensuite, STAVIZ est évalué suivant une méthodologie proposée et des pistes d'amélioration sont données. Finalement, ce travail espère préparer le terrain pour des travaux futurs et fournir des lignes directrices utiles à ceux qui entreprendraient le développement d'un outil pour les analystes visés.

Acknowledgements

I would like first to thank my thesis supervisor Pr. Bruno Dumas and my thesis co-supervisor Pr. Anne Wallemacq from the University of Namur. The door to their office was always open when I faced difficulties and when I needed to discuss ideas. They allowed me to enjoy a great autonomy but they were always there for me when I needed directions and advice.

I would also like to thank the EFFaTA-MeM project members (Pr. Bruno Dumas, Pr. Anne Wallemacq, Pr. Isabelle Linden, and Pr. Guy Deville) for welcoming me among them in the context of my internship and for their great availability. They had a strongly open-minded attitude and were always attuned to suggestions. Special thanks go to Pr. Anne Wallemacq who was my internship mentor for her unfailing supervision.
I also acknowledge Christian Colot, whose office I shared throughout my internship. His friendliness and our conversations contributed to make my internship an unforgettable experience.

Furthermore, I would like to thank my fellow students Thomas, Florian, Pierre, Anne, Jérémy, and Jonathan for their support and because suffering together is more enjoyable than suffering alone.

Finally, I express my gratitude to my parents for the support and the love they gave me throughout my studies. The accomplishment of this thesis would not have been possible without them.

Contents

Chapter 1

Introduction

The tremendous and always increasing availability of raw textual data has created a strong need for text visualization techniques in the past decades. As a result, more and more articles presenting novel visualization techniques are published by researchers worldwide. The Text Visualization Browser[1] (Kucher et Kerren, 2015) lists 380 techniques, 311 of which dating from the last ten years. This phenomenon has such an extent that recent works provide taxonomies and overviews of the existing research ((Šilić et Bašić, 2010), (Alencar *et al.*, 2012), (Liu *et al.*, 2014), (Gan *et al.*, 2014), (Kucher et Kerren, 2015), (Cao et Cui, 2016)). It is undoubtedly supported by the availability of recent free-access toolkits easing the development of such techniques. The D3.js library (Bostock *et al.*, 2011) is a fashionable example.

However, extracting knowledge by looking at a plain text is tedious for humans. This is where text analysis steps in. Different approaches to text analysis exist, the choice of one or of a mix should depend on the questions to answer about the text. Overall, two main approaches exist : quantitative analysis and qualitative analysis (Graneheim et Lundman, 2004).

In 2010, James Hertog (Professor at the University of Kentucky) writes : *"A number of scholars say you cannot capture the meaning of a text by counting the number of times violence is portrayed or the categories of jobs named in a story, etc."* (Hertog, 2010). His quote illustrates that quantitative analysis has limitations regarding the analysis of the meaning of a text.

The text analysis technique of interest in this thesis belongs to the qualitative techniques category. It is called the structural analysis. The objective of a structural analyst is to understand the implied and the connotative meanings that hide in a text. In doing so, it uses the relationship between two terms as the unit of analysis (Piret *et al.*, 1996). The frequency of a term is of no interest, nor is its semantic meaning. The idea of the structural approach is that writing down such relationships will allow discovering implicit ones, that will uncover the true meaning of the text.

[1] http://textvis.lnu.se/ (accessed September 14, 2016)

The value of the structural approach was well illustrated by Critchfield in (Critchfield, 2017). Critchfield tells the story of a behavior specialist that speaks with the parents of one of its patients. Critchfield illustrates that the fact that the specialist expresses himself in jargon involves that he does not have the same perception of some words as his patient's parents do. As a result, his discourse is misinterpreted. For example, the word *extinction* refers to a treatment for behavioral analysts, but it has a negative connotation for non-experts. The structural analysis takes into account the pre-existing perceptions and is able to unravel such misunderstandings.

From a discourse about perceptions of workers from a crisis management center to a comparison of the perception of archery in different cultures, the possible application range of structural analysis is highly broad. Numerous examples are given in (Piret *et al.*, 1996), (Wallemacq et Jacques, 2001), and (Wallemacq *et al.*, 2004).

The objective of this thesis is to bridge the gap between structural analysis and visualization. More precisely, the problematic of interest is to understand how text visualization techniques can help structural analysts in their work.
The general research question of this thesis is the following :

> How can visualization techniques lend a hand to structural analysts?

In order to organize the work into well-defined steps it was broken down into two more specific research questions :

- Which existing visualization techniques and tools could be relevant to structural analysis?
- What makes a visualization tool suitable to help structural analysts?

The work was organized in two stages, each covered by a semester of the 2016-2017 academic year.
The first stage took place from September 2016 to December 2016. It consisted of a three-month internship at the University of Namur under the supervision of Pr. Anne Wallemacq. The objective was to develop a software tool to explore the use of visualization techniques for structural analysis. In doing so, another objective was the familiarization with the key concepts of structural analysis.
The second stage took place from January 2017 to May 2017. The first objective was to explore the existing visualization techniques and tools to determine the ones that could prove relevant to structural analysis. The second objective was to conduct an evaluation of the software solution proposed at the end of the first stage. This evaluation was based on the feedback received on the proposed solution and on a more extensive research about text visualization techniques and structural analysis. The improvements proposed were not implemented in the solution but were used to produce a paper prototype of an example of improved version of STAVIZ.

This thesis is organized as follows. Chapter 2 is a state of the art of potentially relevant visualization techniques. Chapter 3 presents STAVIZ, the aforementioned software solution proposed. More precisely, the technological choices and the functionality are detailed in this chapter. Chapter 4 proposes an evaluation grid to formalize the criteria that a visualization technique must satisfy to be helpful to a structural analyst. Improvements of STAVIZ are then proposed in the same chapter based on the evaluation conducted following this grid. Chapter 5 concludes the thesis by recalling the contributions and the limitations of the work, as well as possibilities of future work.

Chapter 2

State of the Art

This chapter aims at presenting an overview of the existing visualization techniques that are relevant to the present thesis. It is organized into four sections following a coarse-fine granularity approach.

First, four information visualization pipelines are presented. They give a coarse-grained overview of the process that derives a visualization from raw data.
Second, visualization techniques taxonomies from the literature are set forth. This part details the elements of the aforementioned pipelines such as the raw data types, the visualization types, and the interaction types. A taxonomy is then proposed with the purpose to put the presented taxonomies together. It is used to define a filter to focus the scope on the relevant categories of visualization techniques.
Third, this filter is applied to the most up-to-date visualization techniques collection. Eight techniques (seven from the collection) are then briefly illustrated and explained.
Fourth, four techniques are reviewed more extensively and compared with each other. These were implemented in the software solution proposed in the present thesis.

Many other techniques are not mentioned in the subsequent review. The tremendous amount of literature about text visualization and the fast evolution of the field (155 new techniques between 2014 and April 2017[1] (Kucher et Kerren, 2015)) makes it tedious to provide a comprehensive overview.

2.1 Information Visualization Pipeline

The goal of text visualization techniques is to represent raw textual data in a way that will allow an analyst to efficiently gain insight. This process of transforming textual data into a visual and, in most cases, interactive representation is not straightforward. Some researchers have proposed a pipeline to describe it step by step.

[1] http://textvis.lnu.se/ (accessed September 14, 2016)

In 1999, Tan presented a pipeline explaining that the raw text has to be processed before a visualization can be generated (Tan *et al.*, 1999).

In 2000, Chi proposed a pipeline containing four consecutive data stages, namely Value, Analytical Abstraction, Visualization Abstraction, and View (Chi, 2000). The Value is the textual data, the Analytical Abstraction consists of metadata, the Visualization Abstraction is the visualizable information and the view is what the user sees and interacts with. The three former data stages are reachable by applying a transformation to the previous stage.

More recently, Liu et al. presented a 5-stage visualization pipeline that maps raw data to an interactive rendering (Liu *et al.*, 2014). The first stage consists of data pre-processing activities such as the extraction of structured data and the removal of noise. The second stage filters data. The third stage maps the pre-processed filtered data to visualizable data. The fourth stage renders the output of the previous stage. Finally, the fifth stage allows interaction with the rendering.

Overall, the three pipelines mentioned above show that a visualization cannot be generated directly from raw data and that a pre-processing step is necessary. The general information visualization pipeline could be summarized as shown in Figure 2.1.

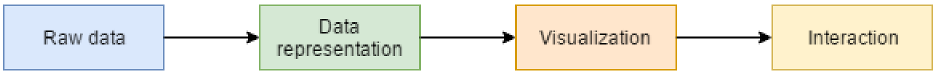


Figure 2.1: General Information Visualization Pipeline

Various works have been published to extend the infovis pipeline. For example, Jansen and Dragicevic have proposed a pipeline which elaborates how the user gets information from a representation (Jansen et Dragicevic, 2013). More precisely, they explain that a user can get multiple perspectives from a representation and combine them to form a mental visual model of what they see. Finally, this mental visual model can be understood to get an insight of the raw data. Figure 2.2 shows the infovis pipeline, as extended by Jansen and Dragicevic.

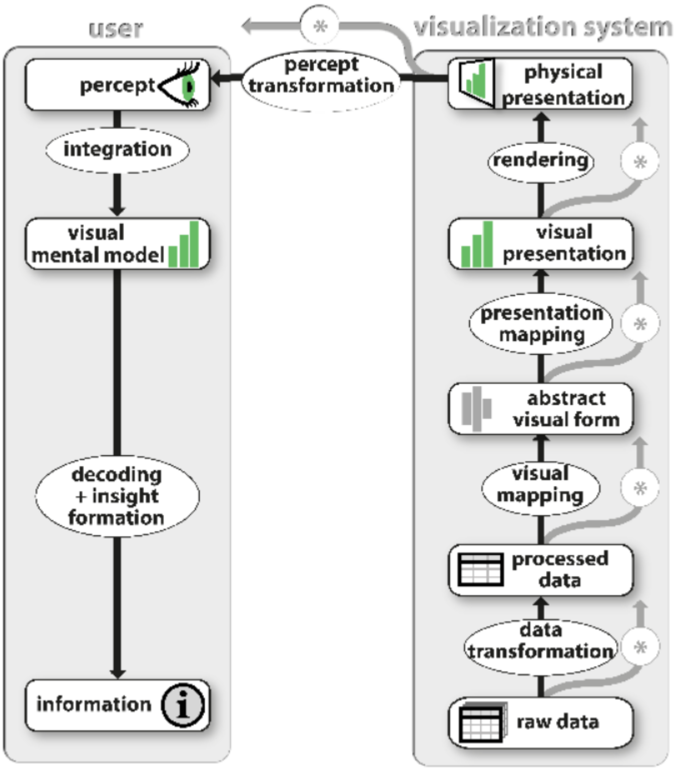


Figure 2.2: Jansen and Dragicevic extended infovis pipeline (Taken from (Jansen et Dragicevic, 2013))

2.2 Existing Taxonomies

Many taxonomies have been proposed in order to classify the techniques for each part of the pipeline. While some of them focus only on one element of the pipeline, others combine multiple elements.

2.2.1 General Infovis Taxonomies

In 1996, Shneiderman expresses the Information Seeking Mantra: *"Overview first, zoom and filter, then details on demand"* (Shneiderman, 1996). Based on this fundamental guideline, he proposes an information visualization taxonomy describing seven data types and seven interaction tasks. Since only textual data is of interest, only the tasks are mentioned. These tasks are overview, zoom, filter, details-on-demand, relate, history, and extract.

Four years later, Chi proposed a taxonomy of visualization techniques based on his aforementioned pipeline (Chi, 2000). He detailed the data stages and the transformations applied between the stages for 36 visualization techniques. He argues that breaking each technique in this way improves reusability when designing new ones.

In 2002, Keim presented a taxonomy that combines three dimensions, namely the data to be visualized, the visualization technique and the interaction/distortion technique (Keim, 2002). The "data to be visualized" part is not detailed since textual data is only one possible value of this dimension. The visualization techniques are standard 2D/3D display, geometrically-transformed display (for instance, projections), iconic display, dense pixel display, and stacked display. The interaction/distortion techniques are dynamic projection, interactive filtering, interactive zooming, interactive distortion, and linking-and-brushing. Keim argues that the three dimensions of his classification are orthogonal.

In 2005, Amar, Eagan, and Stasko proposed a taxonomy that focuses on users' goals (Amar *et al.*, 2005). Their classification consists of ten analytic tasks determined from users' questions regarding real data sets: retrieve value, filter, compute a derived value, find an extremum, sort, determine a range, characterize a distribution, find anomalies, cluster, and correlate.

Two years later, Yi et al. presented a taxonomy of interaction techniques (Yi *et al.*, 2007). They argue that some interaction techniques classifications are system-centric such as (Shneiderman, 1996) and (Keim, 2002) while their and others (for instance, (Amar *et al.*, 2005)) focus on users' goals. The goals featured in their taxonomy are: select, explore, reconfigure, encode, abstract/elaborate, filter, and connect.

2.2.2 Text Visualization Techniques Taxonomies

In 2010, Šilić separated the information visualization process into three steps that match the general information visualization pipeline (Figure 2.1) (Šilić et Bašić, 2010). He describes the textual data types, the intermediate representations of text, as well as the different approaches to draw a view and to provide interaction. Figure 2.3 summarizes Šilić's taxonomy.

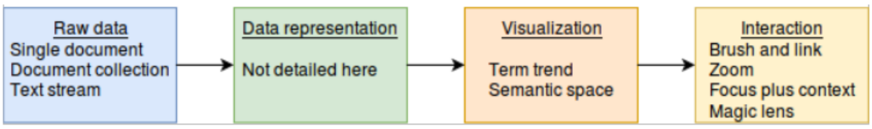
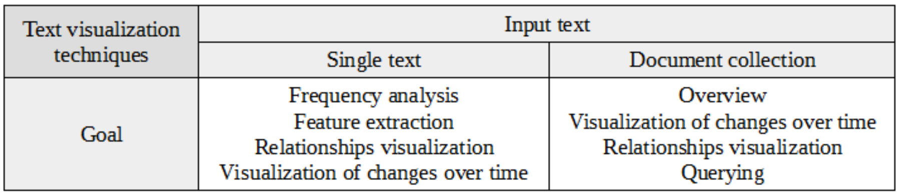


Figure 2.3: Šilić's taxonomy

In 2012, Alencar, de Oliveira, and Paulovich proposed a 2-dimensional classification of text visualization techniques (Alencar *et al.*, 2012). These dimensions are the input text and the goal of the technique. The techniques are first separated based on the input text (either a single text or a document collection) and then based on the goal. Figure 2.4 summarizes the authors' taxonomy.



Text visualization techniques	Input text	
	Single text	Document collection
Goal	Frequency analysis Feature extraction Relationships visualization Visualization of changes over time	Overview Visualization of changes over time Relationships visualization Querying

Figure 2.4: Alencar's, de Oliveira's and Paulovich's taxonomy

Along with their information visualization pipeline, Liu et al. presented a taxonomy of the research in information visualization in (Liu *et al.*, 2014). It is organized into four categories: empirical methodologies (namely, theoretical foundations), interactions, frameworks, and applications. Only the interaction and the application categories are of interest in the present case. The authors distinguish WIMP and post-WIMP interactions. The application represents the visualized data type, textual data corresponds to the second category of applications. Figure 2.5 shows a further decomposition of text visualization techniques.

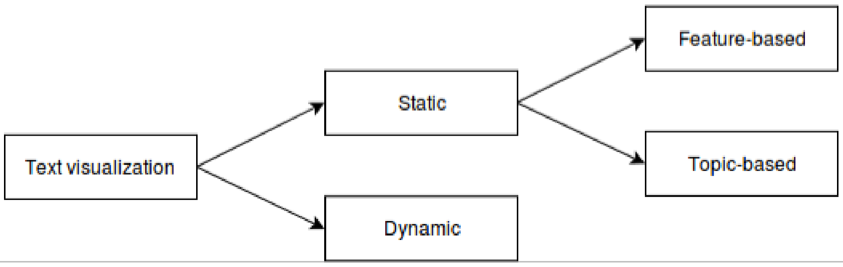


Figure 2.5: Liu et al. decomposition of text visualization techniques

The same year, Gan et al. proposed another classification of text visualization techniques (Gan *et al.*, 2014). In their taxonomy the techniques are first organized into three categories, according to their text type and to their application domain versatility. These categories are techniques for single documents, techniques for document collections and domain-specific techniques. The two former are not domain-specific. A deeper classification is shown in Figure 2.6. Moreover, an interesting feature about their taxonomy is that the visualization tools they mention as examples are evaluated according to the seven tasks in (Shneiderman, 1996).

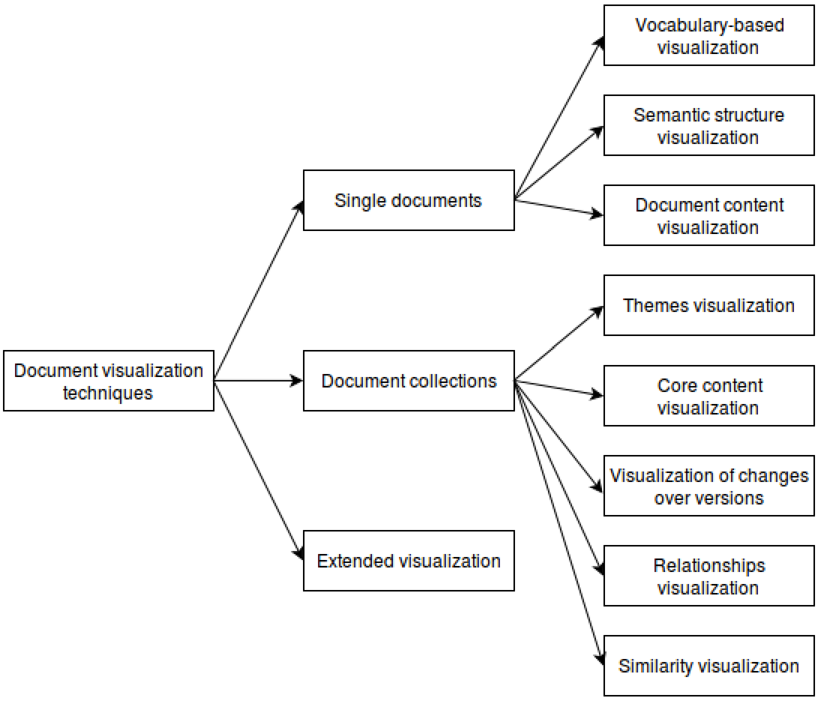

Vocabulary-based visualization
Semantic structure visualization
Document content visualization
Single documents
Themes visualization
Document visualization techniques
Document collections
Core content visualization
Visualization of changes over versions
Extended visualization
Relationships visualization
Similarity visualization

Figure 2.6: Gan et al.'s taxonomy

In 2015, Kucher and Kerren presented a fine-grained taxonomy of text visualization techniques (Kucher et Kerren, 2015). It currently gathers 380 techniques (last updated on April 24, 2017), as compared to the 141 initially listed. It is considered as the most complete taxonomy there is in text visualization (Cao et Cui, 2016). Their work is available online as an interactive browser (Text Visualization Browser, abridged TVB) that allows seeing a thumbnail of all the techniques and filtering them according to the dimensions of the taxonomy[2]. Figure 2.7 shows these dimensions in detail.

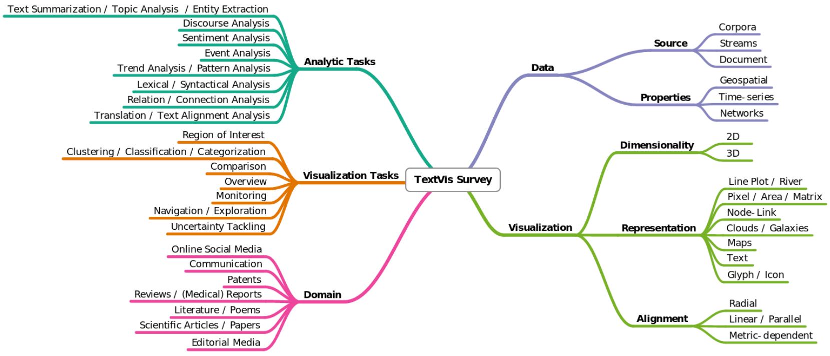


Figure 2.7: Kucher's and Kenner's taxonomy (taken from (Kucher et Kerren, 2015))

[2] www.textvis.lnu.se (accessed September 14, 2016)

Recently, Cao and Cui proposed a text visualization techniques taxonomy that groups them into four categories according to their design goals (Cao et Cui, 2016). The four categories gather techniques for respectively visualizing document similarity, revealing content, visualizing sentiments and emotions, and exploring document collections. Figure 2.8 gives the complete classification.

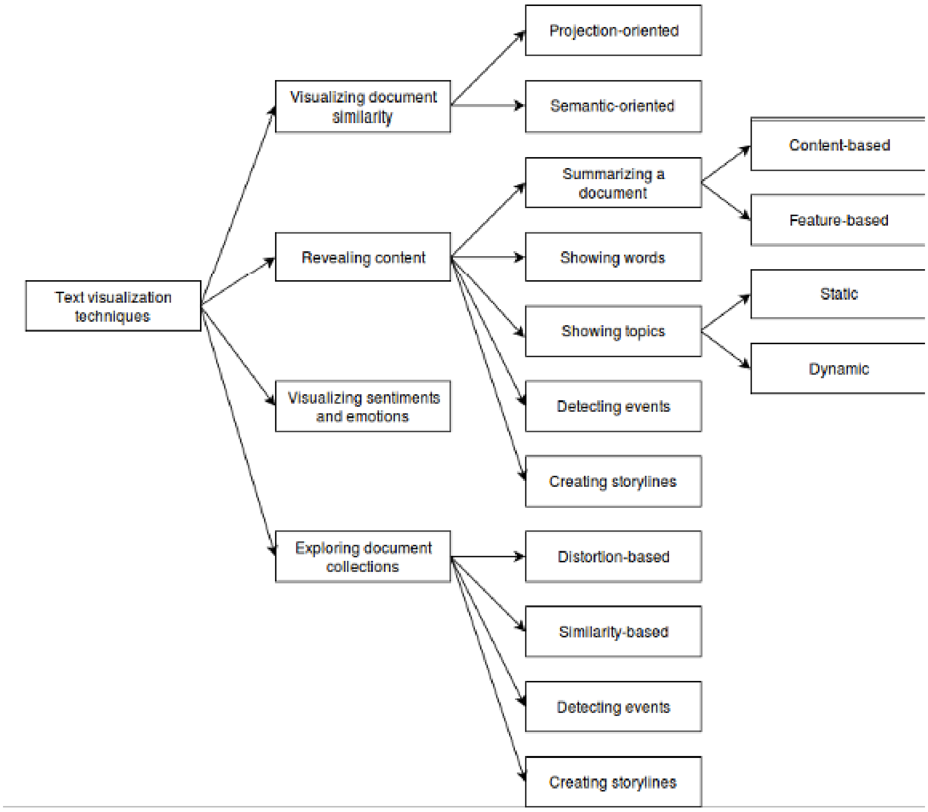


Figure 2.8: Cao's and Cui's taxonomy

Overall, several authors use the *single text or document collection* criterion as a starting point for their taxonomy ((Šilić et Bašić, 2010), (Alencar *et al.*, 2012), (Gan *et al.*, 2014)). The distinction between static and time-dependent textual data is also a popular approach, especially for document collections. However, (Šilić et Bašić, 2010) shows that a single text can be considered as a document collection by considering its paragraphs as distinct texts.

Furthermore, taxonomies based on higher-level tasks or goals seem to achieve better partitions of text visualization techniques. The aforementioned evaluation of techniques in (Gan *et al.*, 2014) illustrates that they often fall into several categories when using the tasks in (Shneiderman, 1996) as criterion.

2.3 Existing Text Visualization Techniques

With the purpose of getting a simple yet broad overview of what exists in the field of text visualization, the following general taxonomy is proposed. It attempts to provide a simplified merging of the recent aforementioned taxonomies and to classify the visualization techniques according to what users want to do with them. The general text visualization techniques taxonomy is shown in Figure 2.9.

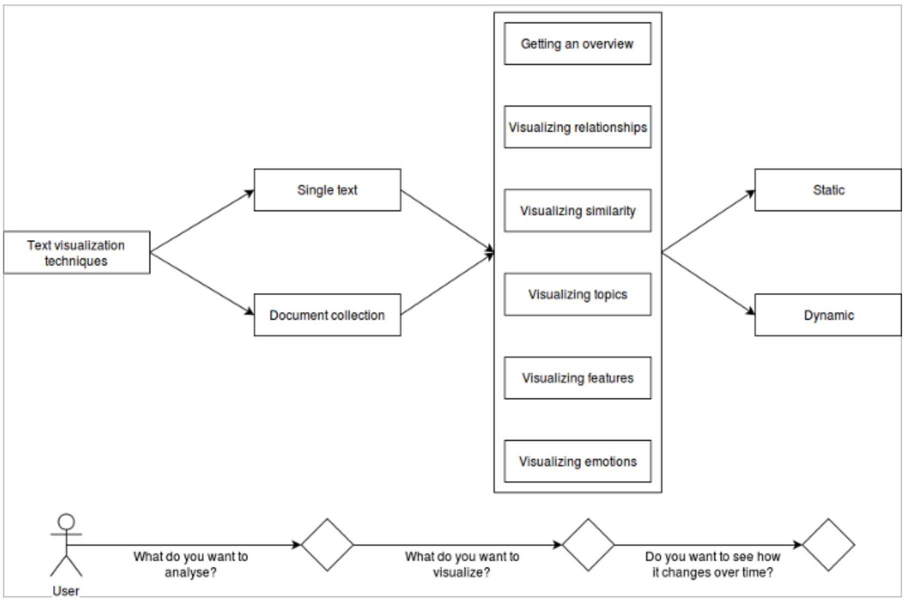

Text visualization techniques
Single text
Document collection
Getting an overview
Visualizing relationships
Visualizing similarity
Visualizing topics
Visualizing features
Visualizing emotions
Static
Dynamic
User
What do you want to analyse?
What do you want to visualize?
Do you want to see how it changes over time?

Figure 2.9: General text visualization techniques taxonomy

The taxonomy's categories are :

- Visualizing the word distribution : show the frequency of the words in a text or in a document collection
- Visualizing relationships : show the links between entities (words, phrases, documents)
- Visualizing similarity : group together entities that are similar according to some criterion (for example, articles that deal with similar topics)
- Visualize topics : detect and show the main topics and possibly their evolution in a text or in a document collection
- Visualize features : show information other than content such as the author or the phrase length about a text or a document collection
- Visualize emotions : show information about the feelings of the author(s) of the text or the document collection

Since this thesis focuses on the structural analysis, some categories of the general taxonomy are of lesser interest here. The taxonomy proposes three questions a user can answer to see which techniques are relevant to their problem. In the case of an analyst that applies structural analysis, the answers would be :

- What does the user want to analyze? : a single text
- What does the user want to visualize? : relationships between words or groups of words
- Does the user want to visualize the evolution over time? : maybe, both static and dynamic views could be interesting

2.3.1 Overview

An interesting way to get an overview of the relevant existing techniques is to apply filters on the TVB. Figure 2.10 shows the results returned after applying two filters : (1) filter on the analytic task : keep only the techniques that can be used to analyze relations and connections (2) filter on the data source : keep only the "Document" data source.

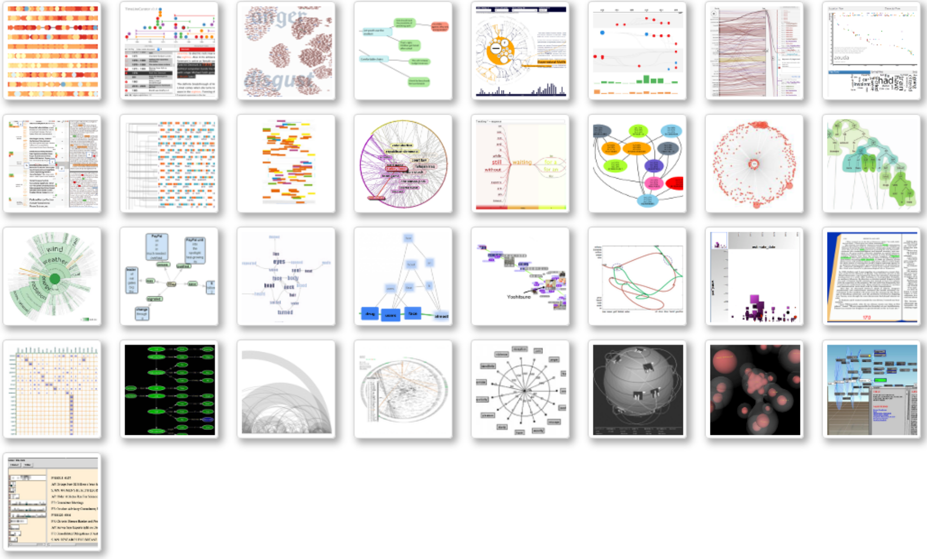
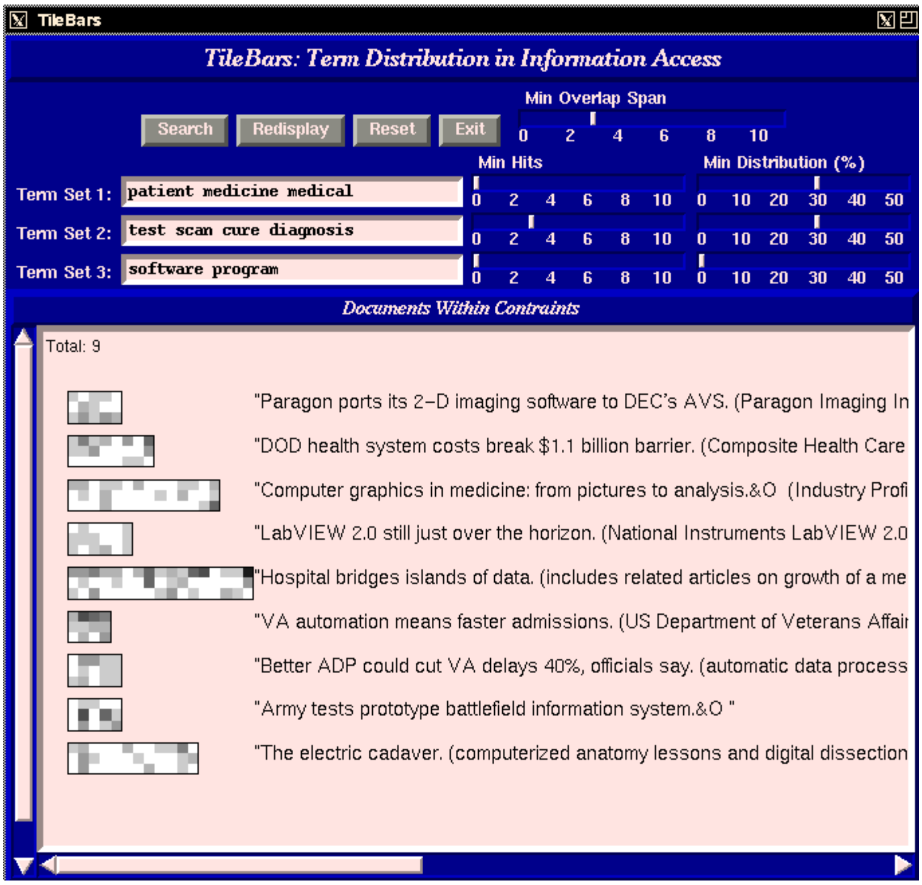


Figure 2.10: TVB : filtering on the "relations and connections analysis" task and on the "document" data source (33 techniques are displayed)

Seven techniques from this list as well as an additional one are described below.

In 1995, Hearst introduced TileBars, a technique for visualizing a set of documents returned as a result of a keyword-based user query (Hearst, 1995). TileBars represents a document as a rectangle horizontally divided into squares depicting non-overlapping parts of the document referred to as tiles. When the user enters a characters string, TileBars returns the documents where the string appears the most. The tiles of the documents are colored in gray if they contain an occurrence of the string, with the darker tiles holding the most occurrences. As shown in Figure 2.11, users can see the query results for several keywords at one time. This allows spotting interesting co-occurrence relationships between the searched words.



Term Set 1: patient medicine medical

Term Set 2: test scan cure diagnosis

Term Set 3: software program

Documents Within Contraints

Total: 9

"Paragon ports its 2-D imaging software to DEC's AVS. (Paragon Imaging In

"DOD health system costs break $1.1 billion barrier. (Composite Health Care

"Computer graphics in medicine: from pictures to analysis.&O (Industry Profi

"LabVIEW 2.0 still just over the horizon. (National Instruments LabVIEW 2.0

"Hospital bridges islands of data. (includes related articles on growth of a me

"VA automation means faster admissions. (US Department of Veterans Affair

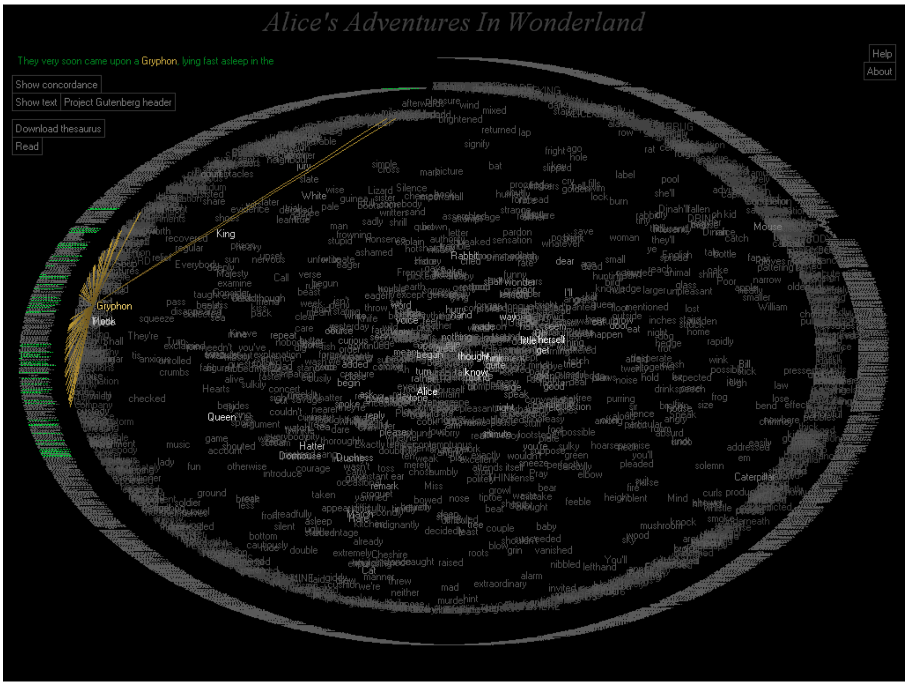
"Better ADP could cut VA delays 40%, officials say. (automatic data process

"Army tests prototype battlefield information system.&O "

"The electric cadaver. (computerized anatomy lessons and digital dissection

Figure 2.11: TileBars example : there are three layers of tiles for each returned document, one for each query (taken from (Hearst, 1995))

TextArc was presented in 2002 by Paley (Paley, 2002). It is drawn in two steps: (1) the text is written line by line on an ellipse and (2) a bag of words of the text is generated (stopwords are removed and words are stemmed) and written inside the ellipse. The position of the word is the centroid of the lines where there is an occurrence of the word. Most frequent words appear brighter to be more visible. Figure 2.12 shows the TextArc generated for Lewis Carroll's novel *Alice's Adventures in Wonderland*. Hovering a word colors all its occurrences on the ellipse in green.



They very soon came upon a Gryphon, lying fast asleep in the

Figure 2.12: TextArc example, the word Gryphon is hovered (taken from http://www.textarc.org/images/alice3.gif)

The same year, Wattenberg presented the Arc Diagram, a technique that shows word repetitions in a text (Wattenberg, 2002). In his work, the analyzed text is written along a horizontal line. The leftmost (resp. rightmost) part of the line corresponds to the beginning (resp. end) of the text. A repetition relationship between two word sequences is represented by an arc connecting two consecutive occurrences of the sequence on the line. The thickness of the arc varies with the length of the repeated sequence. As shown in Figure 2.13, Arc Diagrams can also be used to study repetition patterns in music.

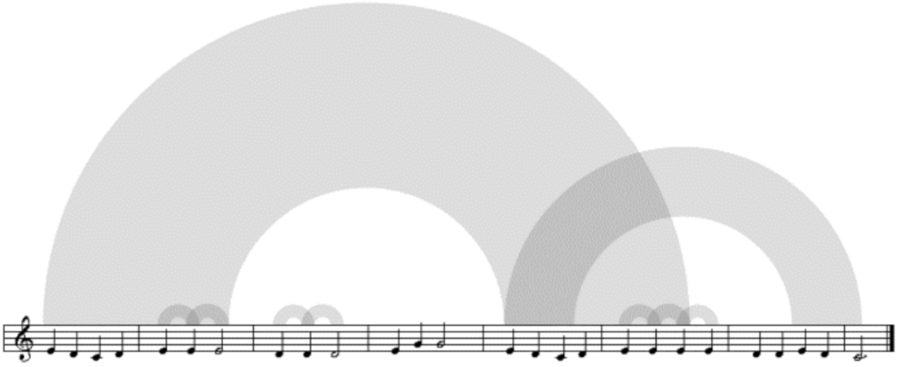
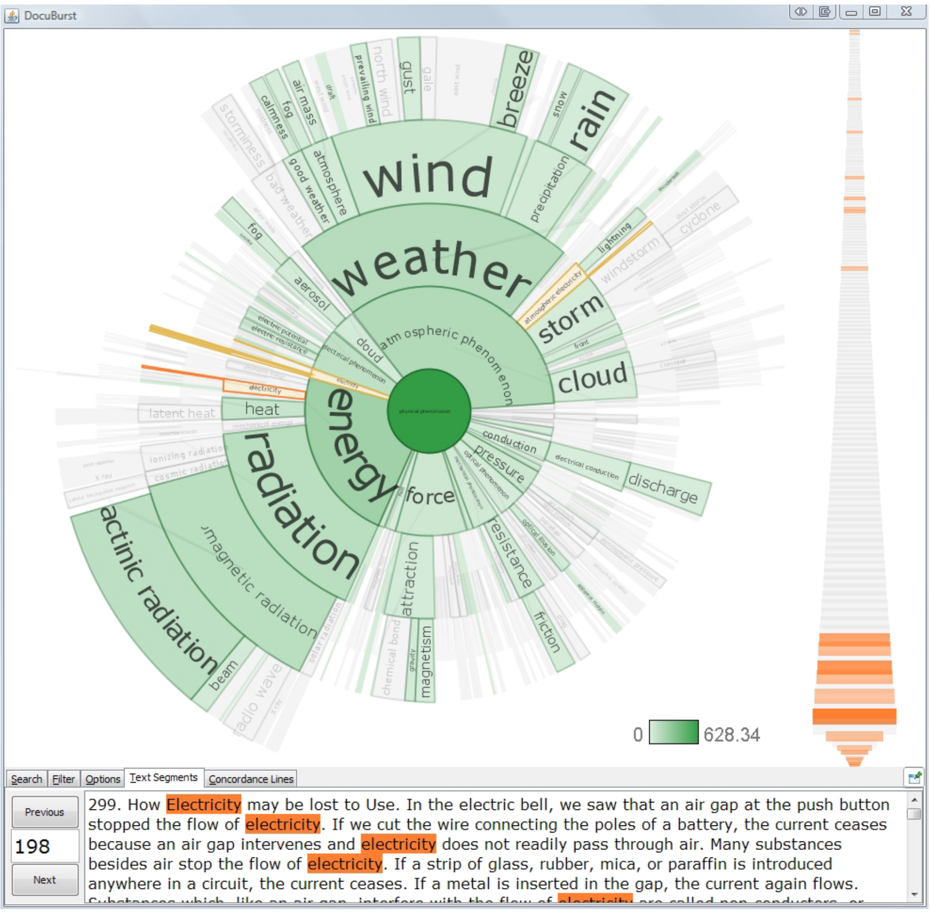


Figure 2.13: Arc Diagram of the first line of *Mary had a little lamb* (taken from (Wattenberg, 2002)

In 2009, Collins, Carpendale and Penn introduced DocuBurst, a sunburst visualization technique that displays hyponymy relationships between words (Collins *et al.*, 2009). These relationships come from WordNet and are not derived from the analyzed text. However, DocuBurst offers interaction tools to observe the distribution of a word in the text. As in the TileBars visualization, the text is depicted as a set of tiles. When a word hovers in the DocuBurst, the tiles are highlighted in a shade that reflects the frequency of the word. The original text with the highlighted occurrences can be displayed by clicking on a tile. Figure 2.14 shows an example taken from (Collins *et al.*, 2009).



299. How Electricity may be lost to Use. In the electric bell, we saw that an air gap at the push button stopped the flow of electricity. If we cut the wire connecting the poles of a battery, the current ceases because an air gap intervenes and electricity does not readily pass through air. Many substances besides air stop the flow of electricity. If a strip of glass, rubber, mica, or paraffin is introduced anywhere in a circuit, the current ceases. If a metal is inserted in the gap, the current again flows. Substances which, like an air gap, interfere with the flow of electricity are called non-conductors, or

Figure 2.14: DocuBurst example : the word ”electricity” is highlighted (taken from (Collins *et al.*, 2009))

The same year, van Ham, Wattenberg and Viégas presented a technique that represents a text as a directed word graph : Phrase Nets (Van Ham *et al.*, 2009). Their work differs from the aforementioned techniques because the unit of analysis is the phrase (thus, the relationship between words) and not the word alone. Also, the Phrase Net represents several types of relationships (e.g. : A and B, A is B) and allows users to define their owns with simplified regular expressions.

The legibility is ensured by two post-processing steps improving the result of the layout algorithm : overlap removal and edge compression. The words on the visualization also convey information. The size (resp. color) represents the frequency of a word in the text (resp. the outdegree to indegree ratio). As for interaction, zooming and filtering possibilities are available.
Figure 2.15 shows an example of Phrase Net. The authors analyzed a set of 7000 British novel titles and generated a visualization with the genitive "A's B" relationship. The Phrase Net shows that there are two main words in their data set (daughter and woman) and that "daughter" tends to be "possessed" (more novels named X's daughter than daughter's X) and that "woman" tends to "possess".

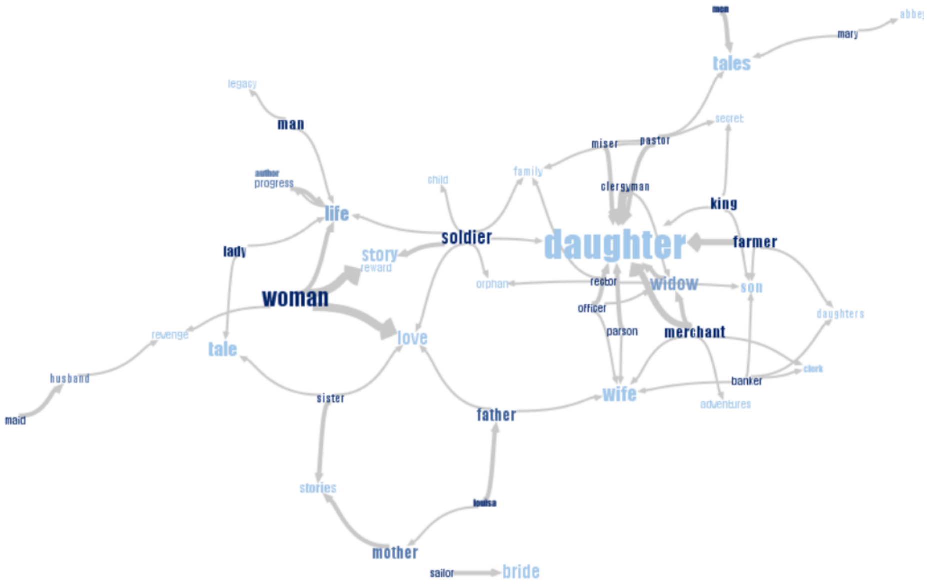
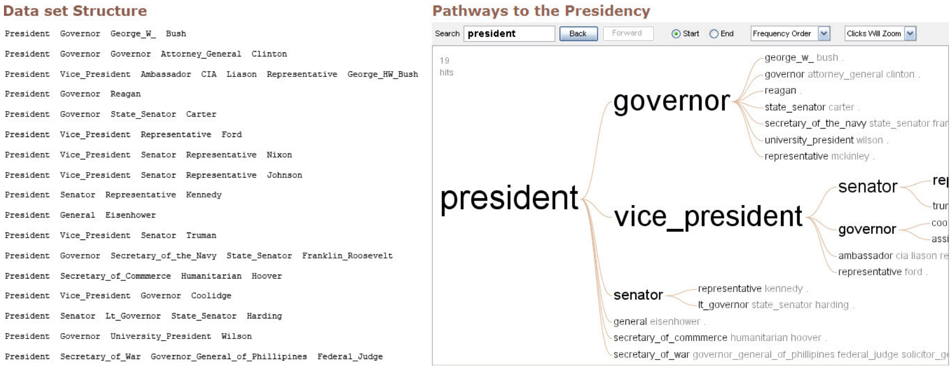


Figure 2.15: PhraseNet example : comparing woman and daughter using the A's B relation pattern (taken from (Van Ham *et al.*, 2009))

Phrase Nets share similarities with the Word Tree introduced in 2008 by Wattenberg and Viégas (Wattenberg et Viégas, 2008). The Word Tree represents the text as a tree where each node is a group of words and where a link between two nodes means that the word groups are consecutive somewhere in the text. The word frequency is represented by the font size and the tree structure allows users to be aware of the context. Users can choose the root node of the tree and they can sort the branches by frequency, by order in the text, and alphabetically. As an example, Figure 2.16 shows the path to the presidency of former US presidents.



Data set Structure

```
President  Governor  George_W_  Bush
President  Governor  Governor  Attorney_General  Clinton
President  Vice_President  Ambassador  CIA  Liason  Representative  George_HW_Bush
President  Governor  Reagan
President  Governor  State_Senator  Carter
President  Vice_President  Representative  Ford
President  Vice_President  Senator  Representative  Nixon
President  Vice_President  Senator  Representative  Johnson
President  Senator  Representative  Kennedy
President  General  Eisenhower
President  Vice_President  Senator  Truman
President  Governor  Secretary_of_the_Navy  State_Senator  Franklin_Roosevelt
President  Secretary_of_Commmerce  Humanitarian  Hoover
President  Vice_President  Governor  Coolidge
President  Senator  Lt_Governor  State_Senator  Harding
President  Governor  University_President  Wilson
President  Secretary_of_War  Governor_General_of_Phillipines  Federal_Judge
```

Pathways to the Presidency

Search
president
Back
Forward
○ Start
○ End
Frequency Order
Clicks Will Zoom
19
hits
president
governor
george_w_ bush .
governor attorney_general clinton .
reagan
state_senator carter .
secretary_of_the_navy state_senator fran
university_president wilson .
representative mckinley .
vice_president
senator
rep
trur
governor
coo
assi
ambassador cia liason re
representative ford .
senator
representative kennedy .
lt_governor state_senator harding .
general eisenhower .
secretary_of_commmerce humanitarian hoover .
secretary_of_war governor_general_of_phillipines federal_judge solicitor_g

Figure 2.16: Word Tree example (taken from (Wattenberg et Viégas, 2008))

The similarity with the Phrase Nets lies in the fact that they both represent, in their own way, relationships between consecutive terms in a text, that is, relationships of the type A x B. The ”x” is depicted as a directed link and is a stopword or a user defined word in the case of Phrase Nets. As for Word Trees, the ”x” is represented by a node or a part of a node in the tree. Figure 2.17 (resp. 2.18) shows how the relationship ”son of God” would be represented in a Word Tree (resp. Phrase Net).

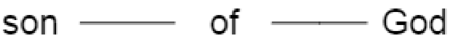


Figure 2.17: The relationship ”son of God” in a Word Tree

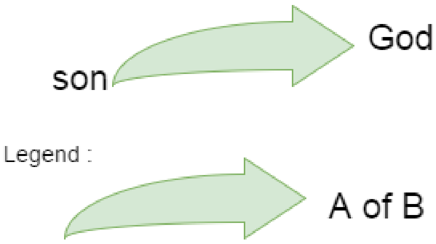


Figure 2.18: The relationship ”son of God” in a Phrase Net

In 2009, Rusu et al. proposed Semantic Graphs, a technique to represent the sentences of a text as a set of relationships in a graph (Rusu *et al.*, 2009). It is similar to the Phrase Net in that it also uses the sentence as the unit of analysis. The relationships discovery process is, however, different. In the Semantic Graph, a sentence is parsed into a triplet (subject, verb, object). These triplets are represented on a node-link diagram in the form subject – verb – object. However, in order to get a meaningful and compact visualization, processing steps are performed before and after the triplets extraction. Figure 2.19 shows an example of Semantic Graph about Ebay.

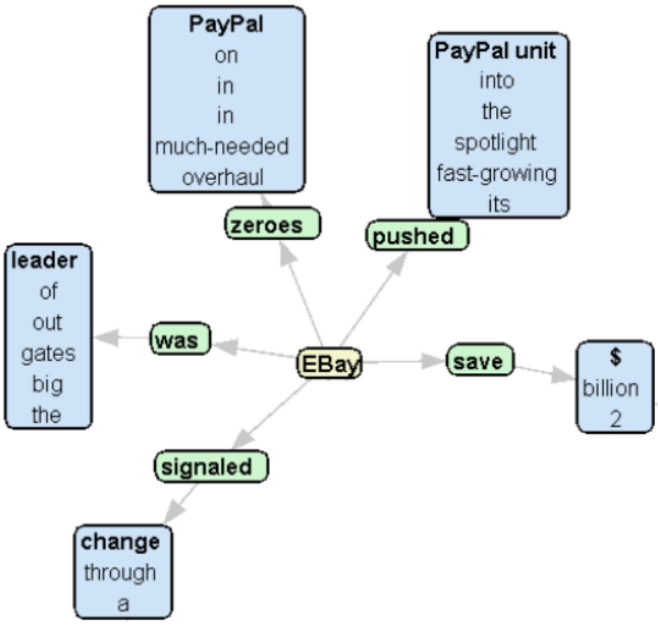


Figure 2.19: Semantic Graph example (taken from (Rusu *et al.*, 2009))

In their work on VarifocalReader (Koch *et al.*, 2014), Koch et al. state that DocuBurst, Phrase Nets and Word Tree allow visualizing relationships, but do not give access to the original text for detailed analysis. However, they argue that abstraction, although necessary to process large documents, is not sufficient for in-depth analysis and that accessing the original text is needed to verify hypotheses. With this in mind, the authors introduced VarifocalReader, a text analysis tool that provides several layers of abstraction (word cloud, topics, source text). Figure 2.20 shows the GUI of VarifocalReader.

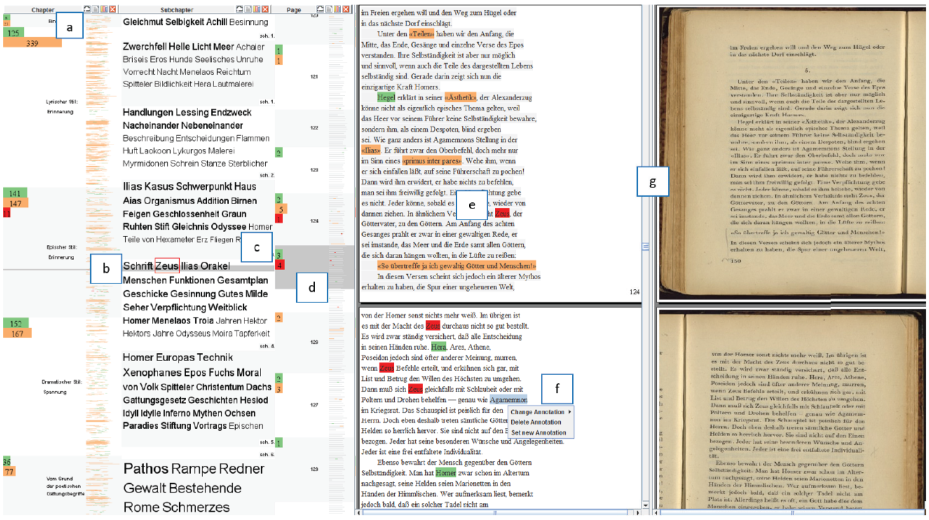

Chapter
Subchapter
Page
a
b
c
d
e
f
g
Gleichmut Selbigkeit Achill Besinnung
Zwerchfell Helle Licht Meer Achaier
Handlungen Lessing Endzweck
Ilias Kasus Schwerpunkt Haus
Schrift Zeus Ilias Orakel
Menschen Funktionen Gesamtplan
Homer Europas Technik
Pathos Rampe Redner
Gewalt Bestehende
Rome Schmerzes
Change Annotation
Delete Annotation
Set new Annotation

Figure 2.20: VarifocalReader GUI (taken from (Koch *et al.*, 2014))

Even if VarifocalReader is not really suited nor designed to represent relationships between text elements, it is relevant to mention it here because it brings an additional point of view, that is the importance of having an access to the source text when performing in-depth analysis.

There are many other text visualization techniques for visualizing relationships between words. An exhaustive review of them is out of the scope of this thesis. The interested reader can refer to the TVB, the reference of the associated paper is given for every mentioned technique.

2.3.2 Text Visualization Techniques Used in STAVIZ

This subsection reviews the visualization techniques used in STAVIZ, namely :

- Word cloud
- Node-link diagram
- Chord diagram
- Adjacency matrix

The word cloud represents a set of words and illustrates their frequency in the analyzed text whereas the three over techniques represent a network of relationships between words. The node-link diagram, the chord diagram, and the adjacency matrix are three different ways to visualize the same data set.

Word Cloud Visualization The word cloud consists of a 2D space where the most frequent words of a text are displayed with a size that reflects their frequency in the text. The word cloud was first introduced in 1976 by psychologist Stanley Milgram (Milgram, 1976). The goal of his work was to understand how Parisians geographically perceive their city. He asked participants to draw a map of Paris by hand and to write on it the elements they could think of. He then examined every map and retained the 50 most frequently mentioned elements. He wrote them on a unique map that reflects the global perception of Paris. The more an element was mentioned by the subjects, the bigger in size it appears on the map. Figure 2.21 shows this map.

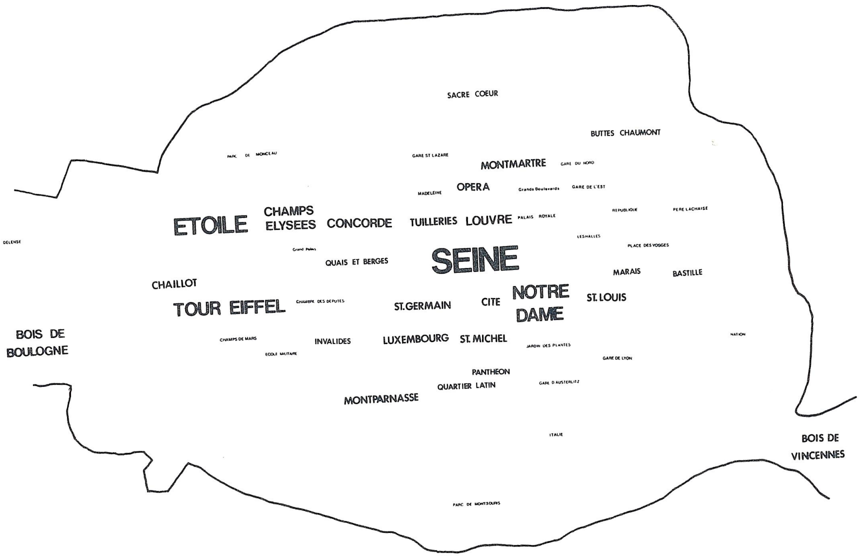


Figure 2.21: The map of Paris in Milgram's study (taken from (Milgram, 1976))

Since then, word clouds have remained a very popular visualization to get a quick overview of a text. Many new techniques attempting to improve the basic word cloud have emerged in the recent years. A search with "word cloud" as keyword in the TVB shows a long yet non exhaustive list of these techniques.

Wordle was proposed in 2009 by Viégas, Wattenberg, and Feinberg (Viegas *et al.*, 2009). They wanted to graphically improve the word cloud by adding colors and by featuring a more pleasing layout. The authors also allow users to customize the rendering. As a result, the users develop a feeling of ownership towards their Wordle, which is extremely important for the authors. Figure 2.22 shows an example of Wordle.

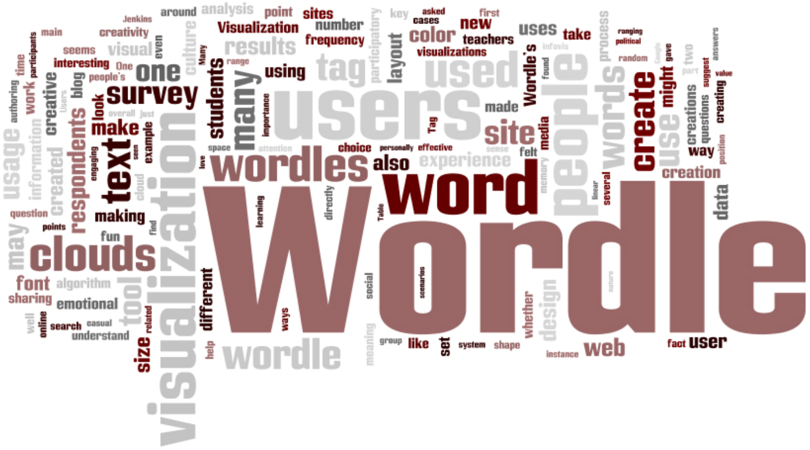


Figure 2.22: Wordle example (taken from (Viegas *et al.*, 2009))

In 2010, Lee, Riche, Karlson and Carpendale presented SparkCloud (Lee *et al.*, 2010). This technique adds the time dimension to the word cloud by showing a frequency graph under every word in the cloud. Figure 2.23 shows an example of SparkCloud.

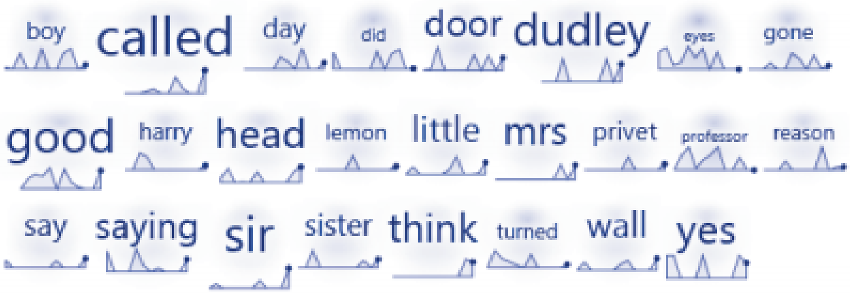


Figure 2.23: SparkCloud example (taken from (Lee *et al.*, 2010))

In 2015, Diakopoulos, Elgesen, Salway, Zhang and Hofland proposed Compare Clouds, a text comparison technique based on word clouds (Diakopoulos *et al.*, 2015). They collected texts from two sources (mainstream media and blogs) and they generated a cloud with all the words that are used in the same context as the word "surveillance". The words that are more mentioned in the source 1 (resp. 2) are redder (resp. bluer) and placed more on the left (resp. right). The words in the middle of the cloud are similarly mentioned in both sources. The vertical axis sorts the words alphabetically. The user can have detail-on-demand by hovering or clicking on a word.

Node-link Diagram Visualization In (Gibson *et al.*, 2013), Gibson et al. define a graph as *"a set of nodes and a set edges such that an edge describes the existence of a relationship between two nodes"*. Graphs are thus perfectly adapted for structural analysis since this technique consists of defining a set of relationships between two words (or groups of words).

A graph is often represented by a node-link diagram (Ghoniem *et al.*, 2005). Figure 2.24 shows an example of the node-link diagram from (Ghoniem *et al.*, 2005). It comprises 50 nodes (represented by a circle with a text label inside) and 400 edges (represented by a line segment connecting two nodes).

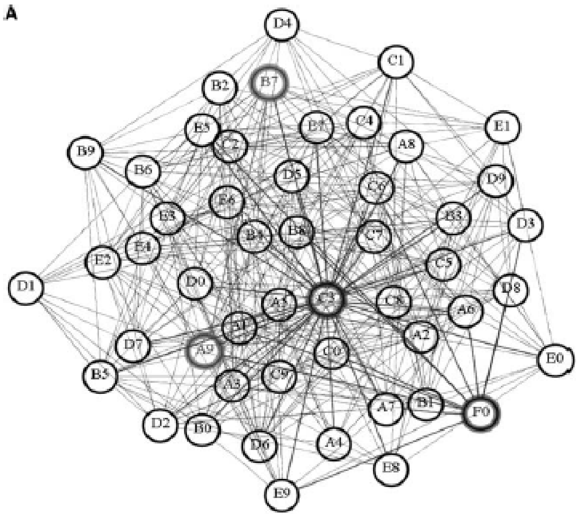


Figure 2.24: Example of node-link diagram (taken from (Ghoniem *et al.*, 2005))

The node-link diagram is generated from the graph by an algorithm. Many algorithms were designed for this purpose, there are consequently many possible node-link layouts for the same graph. A review of the existing work regarding these algorithms is out of the scope of this thesis. The interested reader can refer to (Gibson *et al.*, 2013) for a survey of graph layout techniques. Emphasis will rather be placed on the readability of the node-link representation.

A look at the node-link diagram on Figure 2.24 illustrates that the aesthetic aspect is critical for a user to efficiently analyze the diagram. (Ghoniem *et al.*, 2005) identifies six aesthetic considerations concerning the node-link representation :

- Minimize edge crossings (better legibility)
- Symmetry (better understanding of the structure of the graph)
- Uniform edge lengths (avoid distortion)
- Uniform node distribution (avoid cluttering)
- Separate non-adjacent nodes (close nodes may be though of as connected)
- Node-edge overlap (better legibility)

In 2000, Purchase carried a study to determine the aesthetic factors that impact the performance of the users when they perform analysis tasks on a node-link representation (Purchase, 2000). The factors evaluated in her study are : minimizing bends, minimizing edge crossings, maximizing minimum angles, orthogonality, and symmetry.
Users were asked to answer three questions about ten node-link diagrams. The diagrams were chosen as follows : two diagrams for each factor; one that has a high score for the factor (e.g. a diagram with few edge crossings) and one with

a low score (e.g. a diagram with many edge crossings). The ten diagrams used for her study are shown in Figure 2.25.

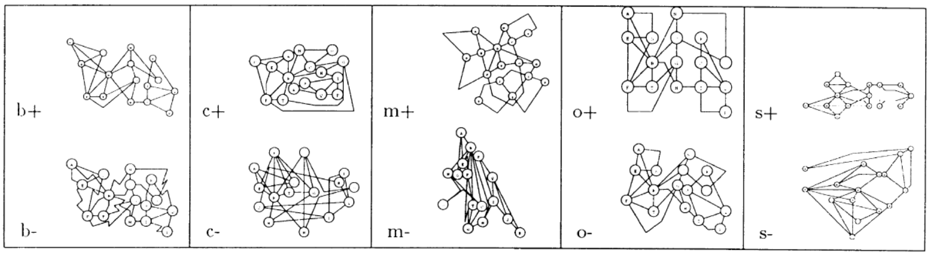


Figure 2.25: Diagrams used for Purchase's study (b = minimizing bends, c = minimizing edge crossings, m = maximizing minimum angles, o = orthogonality, s = symmetry, + = high score, - = low score) (taken from (Purchase, 2000))

Overall, Purchase found that minimizing edge crossings has the greatest impact on the legibility. Symmetry and minimizing edge bending have less impact and maximizing minimum angles and orthogonality have little impact.
However, as shown by Kamada and Kawai in (Kamada et Kawai, 1989), diagrams that minimize edge crossings are not always the best solution. Other factors can play a significant role. Figure 2.26 shows two diagrams taken from (Kamada et Kawai, 1989). The rightmost one has no edge crossings, yet it is deemed more difficult to understand by users.

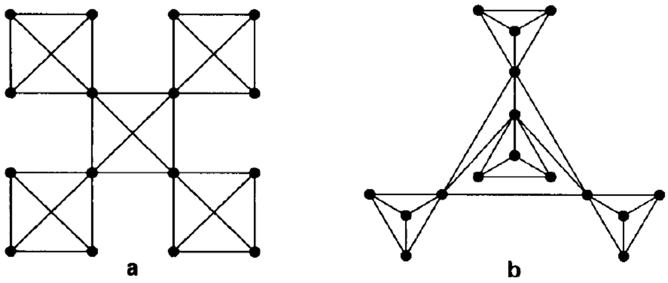


Figure 2.26: Two node-link diagrams - The rightmost one has no edge crossings, yet it is deemed easier to understand by users (taken from (Kamada et Kawai, 1989))

Other work has been published with the purpose of assessing the legibility of the node-link representation. An example (detailed further) is the study conducted by Ghoniem et al. in 2005 (Ghoniem *et al.*, 2005). They also defined user tasks for their assessment.

The work of Lee et al. can serve as a guideline for such an evaluation (Lee *et al.*, 2006). They define a taxonomy of user tasks for node-link diagrams. It is based on the aforementioned taxonomy of Amar, Eagan and Stasko (Amar *et al.*, 2005). Examples of tasks proposed in (Lee *et al.*, 2006) include :

- Find the number of nodes adjacent to a given node

- Find the shortest path between two given nodes
- Find the most connected node

Techniques have been developed to improve the legibility of node-link diagrams. They are especially useful when the graph is too large, which results in visual clutter.

Holten and van Wijk note that providing interaction possibilities (zooming) can help to counter this problem (Holten et Van Wijk, 2009). Furthermore, they proposed an edge bundling technique to tackle this issue. The basic idea of edge bundling is to group together edges that have strong proximity. Figure 2.27 shows an unbundled diagram (leftmost) and a diagram bundled with their technique (rightmost).

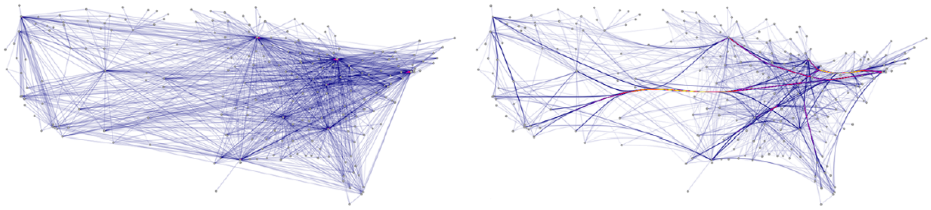
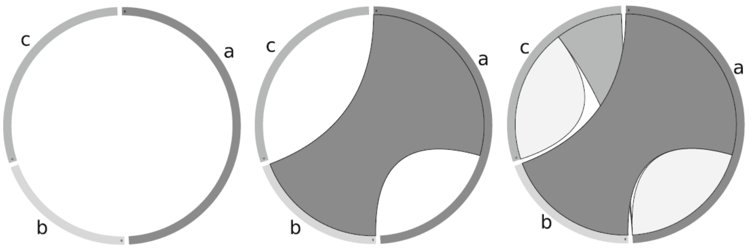


Figure 2.27: Holten and van Wijk edge bundling technique (taken from (Holten et Van Wijk, 2009))

Chord Diagram Visualization A chord diagram is an alternative to the node-link diagram for representing a graph.

In (Jalali, 2016), Jalali defines the chord diagram as a set of arcs and chords. An arc is a portion of the circumference of the circle corresponding to a node and a chord is a portion of the circle connecting two arcs (thus, a chord corresponds to an edge). Figure 2.28 illustrates the elements of a chord diagram.



(a) An example of arcs (b) A chord connecting nodes *a* and *b* (c) A chord diagram

Figure 2.28: Elements of a chord diagram (taken from (Jalali, 2016))

A well-known example of the use of chord diagrams is Circos. It was presented by Krzywinski et al. in (Krzywinski *et al.*, 2009). Circos allows comparing genomes by displaying genomic data in a circular layout. Figure 2.29 gives an example of the abilities of Circos. It shows a comparison between the human and the dog genome. The similarities between the dog and the human genome are represented as chords.

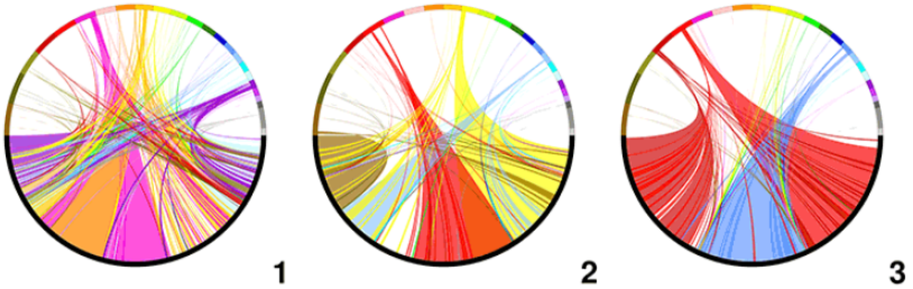


Figure 2.29: Comparison between the human and the dog genome. The lower half (black arc) represents one dog chromosome (a different one for each chord diagram) and the top half (colored arcs) represents the human genome (one chromosome per colored arc) (taken from (Krzywinski *et al.*, 2009))

Another example of chord diagram is the analysis of the co-occurrences of the characters in Victor Hugo's novel *Les Misérables*. A relationship of co-occurrence between two characters is defined when the characters appear in the same chapter. This situation is a graph and can thus be represented by a node-link diagram or by a chord diagram. The two representations are given for comparison (Figure 2.30). The visualizations come from a student project and are available online[3].

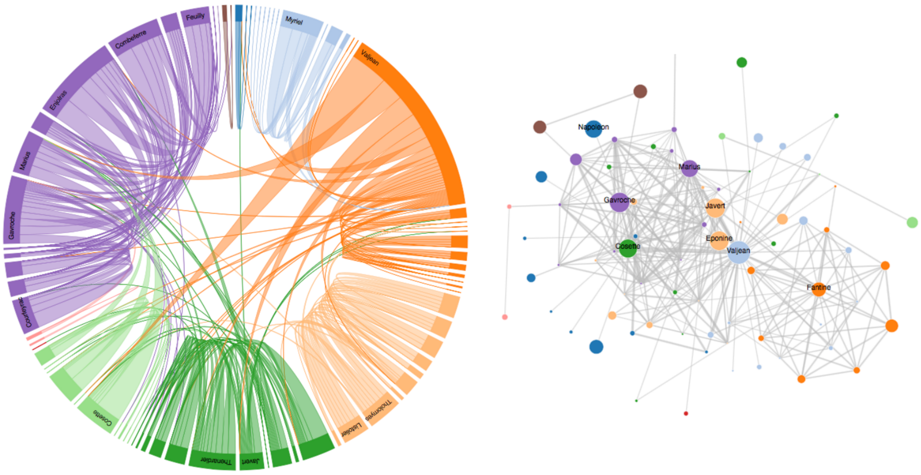
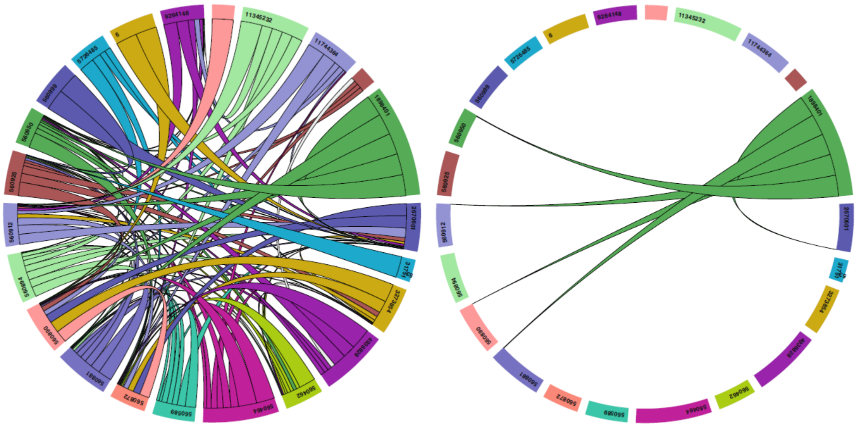


Figure 2.30: Co-occurrence relationships in *Les Misérables* (taken from https://yitianfan.wordpress.com/2014/11/01/network-visualizations/)

The chord diagram gains interest when it becomes possible to interact with it in order to filter relationships. Jalali illustrates this by comparing a non-filtered chord diagram with its filtered version (Figure 2.31). The applied filter shows only the chords connected to a specified arc.

[3] https://yitianfan.wordpress.com/2014/11/01/network-visualizations/ (accessed February 15, 2017)



(a) The complete version

(b) The filtered version based on user interaction

Figure 2.31: Comparison between a non-filtered chord diagram and its filtered version (taken from (Jalali, 2016))

Adjacency Matrix Visualization A graph can be represented by a matrix, named the adjacency matrix associated to the graph. (Ghoniem *et al.*, 2005) defines the matrix visualization of a graph as follows : *"When two vertices are connected, the cell at the intersection of the corresponding row and column contains the value 'true'. Otherwise, it takes on the value 'false'. Boolean values may be replaced with valued attributes associated with the edges that can provide a more informative visualization."*.
In the case of the STAVIZ's matrix, the Boolean values of the above definition are replaced by a color that indicates the type (association or opposition) of the represented relationships.

Figure 2.32 shows the adjacency matrix associated to the node-link diagram in Figure 2.24.

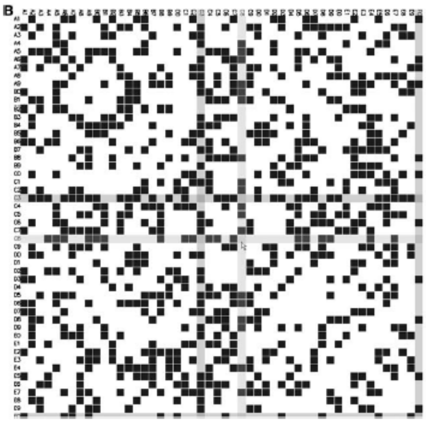


Figure 2.32: Adjacency matrix associated to the node-link diagram on Figure 2.24 (taken from (Ghoniem *et al.*, 2005))

In 2005, Ghoniem, Fekete, and Castagliola compared the readability of the node-link diagram and the adjacency matrix (Ghoniem *et al.*, 2005). They defined seven tasks a user would typically go through when analyzing a node-link diagram or an adjacency matrix and conducted an evaluation to observe how a panel of users performs them. For a given task, the authors wanted to measure the percentage of correct answers, the time taken to answer, and how these numbers were influenced by the size and the density of the graph. Those seven tasks are :

1. Estimate the number of nodes
2. Estimate the number of edges
3. Find the most connected node
4. Find a node given its label
5. Find a link between two given nodes
6. Find a common neighbor given two nodes
7. Find a path between two nodes (of less interest for structural analysis)

Their study shows that the matrix visualization is more adapted than the node-link visualization when analyzing large or dense graphs. This is explained by the fact that the answer correctness percentage and the answering time are much more affected by an increase in size or density in a node-link diagram than in a matrix. The number of nodes sensitivity of the node-link diagram is supported by research scientist Robert Kosara[4].

The adjacency matrix achieves better correctness percentage than the node-link visualization for tasks 1 to 5. As for task 6, the node-link visualization leads for small graphs, but the matrix tends to equalize it for larger graphs.

The study of Ghoniem et al. thus shows that the adjacency matrix has great potential for analyzing a graph and should be exploited.

A great example of the adjacency matrix is Michael Bostock's co-occurrence matrix. It shows the co-occurrence relationships between the characters of Victor Hugo's novel *Les Misérables* (see Figure 2.30 for comparison). It was implemented with d3.js, as Bostock is on the key developers of this library. The matrix visualization is available online[5].

[4] https://eagereyes.org/techniques/graphs-hairball (accessed May 6, 2017)

[5] https://bost.ocks.org/mike/miserables/ (accessed October 20, 2016)

Chapter 3

STAVIZ

This chapter presents the software STAVIZ. It is organized as follows : Section 1 describes the goal of STAVIZ and the context of its development. Section 2 details the technologies used for the development. Section 3 covers the global architecture of STAVIZ. Section 4 details the functionality it offers. Finally, section 5 recalls the contributions of STAVIZ and discusses the methodology.

3.1 Goal and Development Context

STAVIZ is a software that attempts to implement the structural analysis technique. It was developed in the context of a three-month internship at the Faculty of Economics, Social and Management Sciences of the University of Namur.

More precisely, the internship took place in the context of the EFFaTA-MeM research project. This project finds its interest in the analysis and the interpretation of texts. In particular, EFFaTA-MeM focuses on techniques that exploit the whole inner richness of a text whereas more traditional techniques tend to remove any ambiguity from it to make it fully computer-processable. This is why EFFaTA-MeM is interested in structural analysis.

The goal of STAVIZ is to improve the previously developed EVOQ by exploring additional visualization techniques. With this tool in hand, the EFFaTA-MeM project hopes to gather new insight about what makes a text analysis tool genuinely helpful for structural analysis.

3.2 Development Methodology

Before any implementation, the development started with a familiarization with the structural analysis and the previously developed EVOQ. The first week of the internship was dedicated to this step. The remaining time was dedicated to the development of STAVIZ.

The central point of STAVIZ is the visualization functionality. The objective is to provide visualizations that are genuinely helpful, hence the interaction functionality. The two following quotes from Robert Kosara (quoted in (Culy et

Lyding, 2009)) accurately reflect the intention behind the visualizations proposed in STAVIZ :

- *"Nobody wants to look at a table of data, even if it's their own."* : there is a need for visualizations
- *"Visualization has to be more than pretty pictures. It has to challenge. It has to further our understanding. Visualizing data is not about pretty pictures."* : visualizations have to be helpful

The need for adapted visualizations called for an iterative development. A meeting with the members of the EFFaTA-MeM project took place almost every week. The agenda of these meetings was usually a demo, a discussion on the developed functionality, and a planning of the next development iteration.

3.3 Familiarization

The first step of the internship consisted of a familiarization with the structural analysis technique (Piret *et al.*, 1996) and the previously developed EVOQ (Wallemacq *et al.*, 2004). The following subsections summarize the main ideas of structural analysis and EVOQ.

3.3.1 Structural Text Analysis

This subsection is based on the work of Piret, Nizet and Bourgeois (Piret *et al.*, 1996). An extensive presentation of structural analysis is out of the scope of this thesis. All the details, as well as corrected exercises can be found in (Piret *et al.*, 1996).

Definition of Structural Analysis

The structural analysis is a method that aims at highlighting the relationships between the elements of a discourse (words or parts of the discourse, referred to as terms). It seeks to go beyond the explicit content of a text and to show an implicit meaning through these relationships.

The structural analysis belongs to semantic and structural techniques :

- Semantic : the analyst's goal is to grasp the meaning of the discourse
- Structural : the base of the analysis is the relationship between two terms

Disjunction Relationship

The relationship is the unit of analysis. It is a relationship of disjunction, that is, it links two terms having something in common while being different. These two terms refer to (e.g. are connected by) the same semantic axis. They are called the inverse of each other. Figure 3.1 shows a disjunction relationship between the terms "sauvage" *(wild)* and "contrôlée" *(controlled)*. The semantic axis is "les immigrations" *(immigrations)*.

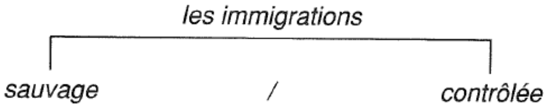


Figure 3.1: Example of disjunction (taken from (Piret *et al.*, 1996))

Moreover, the terms of a disjunction have a positive or negative connotation, which describes how they are globally perceived by the author of the text. The connotation is noted as a valuation index represented by a + (resp. -) sign for a positive (resp. negative) connotation.

The structural analysis gives freedom to the analyst to infer the semantic axis or one of the terms of the disjunction when it is not explicit in the text.

Structures

Disjunction relationships can be combined with others to form more complex structures. The three structures explained in (Piret *et al.*, 1996) are the parallel structure, the hierarchical structure, and the crossed structure.

Parallel Structure A parallel structure is formed by disjunctions which terms are linked two by two by a double implication relationship. Figure 3.2 shows an example of parallel structure.

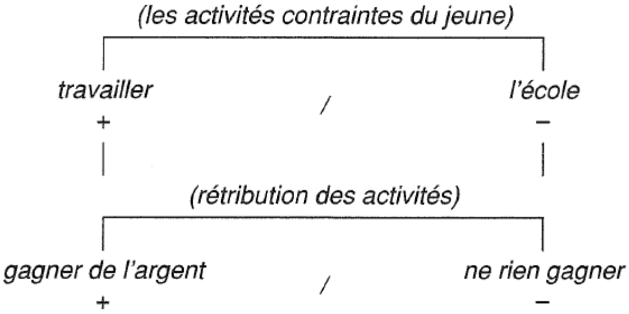

(les activités contraintes du jeune)
travailler
+
/
l'école
−
(rétribution des activités)
gagner de l'argent
+
/
ne rien gagner
−

Figure 3.2: Example of parallel structure (taken from (Piret *et al.*, 1996))

Hierarchical Structure A hierarchical structure can be built when a term is one of the inverses of a disjunction and the semantic axis of another disjunction at the same time. Figure 3.3 shows an example of hierarchical structure about the turntables *(tourne-disques)*. First, the bad (leftmost) turntables are separated from the good ones. Then, the good turntables serve as the semantic axis for a refined classification that divides non-perfected *(pas perfectionné)* turntables from perfected ones.

When a term is the semantic axis of a disjunction, it passes its own valuation index to the terms of this disjunction. In Figure 3.3, the terms *pas perfectionné* and *perfectionné* inherit a + sign in addition to their own valuation index.

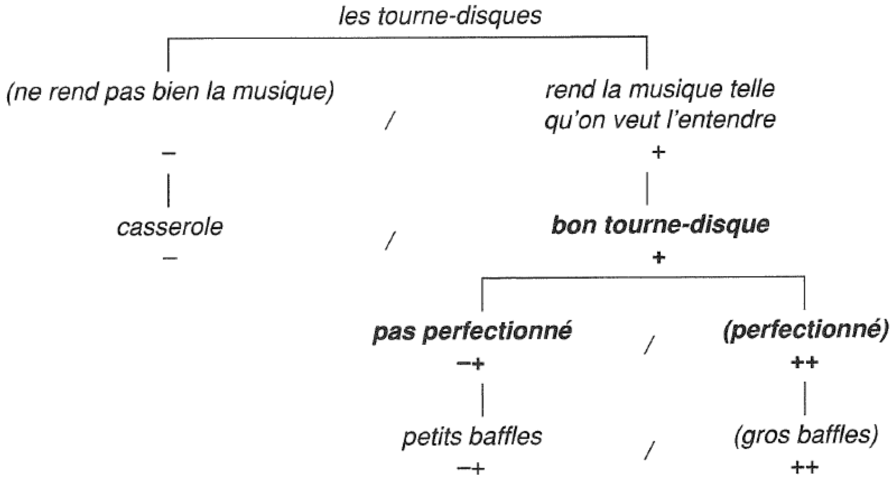


Figure 3.3: Example of hierarchical structure (taken from (Piret *et al.*, 1996))

As illustrated by Figure 3.3, the hierarchical structure can be combined with the parallel structure.

Crossed Structure The crossed structure combines two disjunctions to form four quadrants, each representing a reality, combination of two terms. The two crossed disjunctions are called mother-disjunctions or mother-axis. Figure 3.4 shows an example of crossed structure combining two disjunctions.

The author of the text connotes positively open and enlightened courses. Each reality has a two-sign valuation, inherited from the terms that compose it.

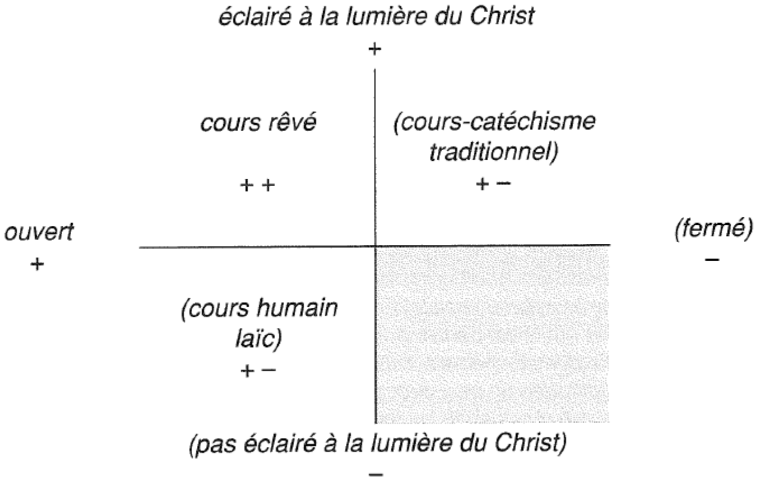

éclairé à la lumière du Christ
+
cours rêvé
+ +
(cours-catéchisme traditionnel)
+ –
ouvert
+
(fermé)
–
(cours humain laïc)
+ –
(pas éclairé à la lumière du Christ)
–

Figure 3.4: Example of crossed structure (taken from (Piret *et al.*, 1996))

However, this structure creates a dilemma when the ++ reality is not accessible. It is impossible to choose between the two +- realities without further information. In this case, the analyst must refer to the text to see if the author solves the dilemma and must add a third valuation index accordingly.

STAVIZ does not use valuation indexes, nor does it specifies any label for the semantic axis. In doing so, the relationships encoding is less tedious.

3.3.2 EVOQ

Objectives of EVOQ and Challenges

EVOQ is an existing cognitive mapping software developed in the context of the EVOQ project (Wallemacq *et al.*, 2004) in 2004. The programming language used for the development was Java 1.4. The objective of this tool is to analyze a text by deriving the semantic fields surrounding its author. In doing so, it uses structural analysis.

The semantic fields represent the author's perceptions. They form the network of the evocations revolving around the terms. The evocations of a term T can be thought of as the terms remembered by T. The meaning (denotation) of a term taken individually is of no interest here, as in the structural analysis. A term is defined by its relationships with other terms.

However, the development of a tool implementing the structural analysis is challenging. The disjunctions are indeed difficult to process for a program because they are not formal, that is, there is no rule to detect all the disjunctions in a text. This is the reason why the disjunctions discovery process cannot be fully automated.

EVOQ Modules

EVOQ has a modular design. It works with four distinct interoperating modules.

Text Module The text module contains the analyzed text.

Dictionary module The dictionary module contains the semantic dictionary, that is, the set of terms and relationships between these terms. A semantic dictionary is bound to a given context.

2D Map Module The 2D map is a 2D dynamic node-link diagram where the nodes are the terms and the links are the relationships between these terms. The visualization in this module is interactive, the user can drag, fix and remove terms. This module also offers a feature allowing users to show the related terms of a term if their choosing. Two terms linked by an disjunction (resp. conjunction) relationship will tend to repel (resp. attract) each other, hence the dynamic aspect of the visualization.

3D Semantic Field Module The 3D semantic field module uses the landscape metaphor to create a 3D field representing the content of the 2D node-link diagram.

Possible Evolutions

(Wallemacq *et al.*, 2004) gives two improvement clues for the dictionary module:

- Add modules to help the analyst with the discovery of relationships
- Explore existing relationships databases such as WordNet

3.4 Technologies

STAVIZ was developed with web technologies, namely HTML, Javascript, and CSS. The reasons for this choice lie in the possibilities offered by these technologies, and mostly by the recent Javascript data visualization library D3.js (Data-Driven Documents). It was the choice of using D3.js that involved the use of HTML, Javascript, and CSS.

D3.js was developed by Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. They presented it in a paper in 2011 (Bostock *et al.*, 2011). D3.js integrates seamlessly with the other web technologies because it shares the same representation of web pages, namely the *document object model.* D3.js can be imported by simply adding the following line to a HTML file. The source can also be downloaded online for free via the official D3 website[1].

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

The strength of D3.js compared to other visualization toolkits is that it binds data directly to the web page element, which allows great control over the visualization (Bostock *et al.*, 2011). This has an advantageous consequence : the visualization doesn't have to be redrawn when a CRUD operation occurs on an element, which enables dynamic visualizations. Furthermore, this increases performance since there is no intermediate representation between the data and the visualization.

In addition, D3.js provides modules that can be reused by developers to solve problems. Examples include the *arc* shape and the *chord* layout which allows generating a layout like the diagrams in (Krzywinski *et al.*, 2009) in few lines of code. The following code snippet (code taken from (Bostock *et al.*, 2011), comments added afterwards) illustrates the usefulness of D3 modules. It works with versions 2 and 3 of D3.js.

```
 d3.select("body").append("svg:svg")
   .data([[1, 1.2, 1.7, 1.5, .7]])
   .attr("width", 150)
   .attr("height", 150)
.selectAll("path")
   .data(d3.layout.pie()) // Use of D3 module (pie layout)
.enter().append("svg:path")
   .attr("transform", "translate(75,75)")
   .attr("d", d3.svg.arc().outerRadius(70)) // Use of D3 module (arc
       shape)
```

[1] https://d3js.org/ (accessed September 15, 2016)

```
// The two following lines were added to the original snippet
  .style("fill", function(d){return getRandomColor();}) // Give a
      random color to each arc (function getRandomColor() not
      predefined)
  .append("svg:title").text(function(d, i){return "Data : " +
      d.value;}); // Display the data on arc hovering
```

This snippet draws a pie chart divided into five arcs (Figure 3.5). With the use of D3 modules, only nine lines of code are needed. The modules used here include the *pie layout* and the *arc* shape.

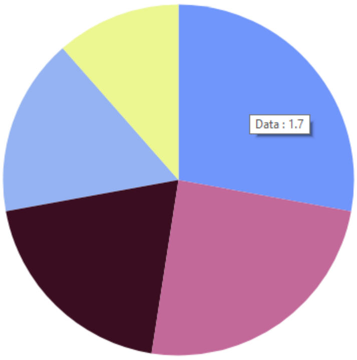


Figure 3.5: Pie chart generated by the modified code snippet from (Bostock *et al.*, 2011)

Drew Skau (PhD. Student in visualization at UNCC) writes *"Perhaps the most important part of D3's success is the position and approach it takes. It is not a graphics library, nor is it a data processing library. It doesn't have pre-built charts that limit creativity. Instead, it has tools that make the connection between data and graphics easy. It sits right between the two, the perfect place for a library meant for data visualization."*[2]. With these words, he illustrates how important the control over the visualization that D3.js offers by the data-element binding is.

Another reason that motivated the choice towards D3.js is its impressively comprehensive documentation and its large users community. The documentation and the variety of examples available online ease the learning of the library and the search for answers to a problem.

D3.js put aside, developing using web technologies has other advantages.
Firstly, web browsers have convenient tools for debugging, which greatly ease the development process. They also offer features such as scroll bars, zooming, local storage, and context menus (accessible via a right click) that can be exploited when developing (e.g. custom context menus) or using (e.g. use the web browser zoom instead of developing a zooming feature) a web application.

[2] http://www.scribblelive.com/blog/2013/01/29/why-d3-js-is-so-great-for-data-visualization/ (accessed April 29, 2017)

Secondly, developing with web technologies allows sharing through the Internet. Since STAVIZ is a text analysis tool, using the Internet as a channel for collaboration between analysts would certainly yield added value. For now, STAVIZ is still in its development stage and reflection is ongoing. It has thus not yet been deployed. Since STAVIZ was developed using web technologies, elaborating an online analysis sharing platform on top of it should not be unduly troublesome.

3.5 Global Architecture

STAVIZ is composed of various files, each of which has a specific responsibility in the global functioning. They can be organized into four categories :

- HTML files
- Javascript files
- CSS files
- External libraries

The external libraries are Javascript files, but they belong to another category because they were not developed in the context of STAVIZ.

HTML Files The HTML files are the web pages accessed by the users when they use STAVIZ. There are six HTML files in total, one for the main page and one for each of the five visualizations. Figure 3.6 shows these six HTML files and the relationships between them.

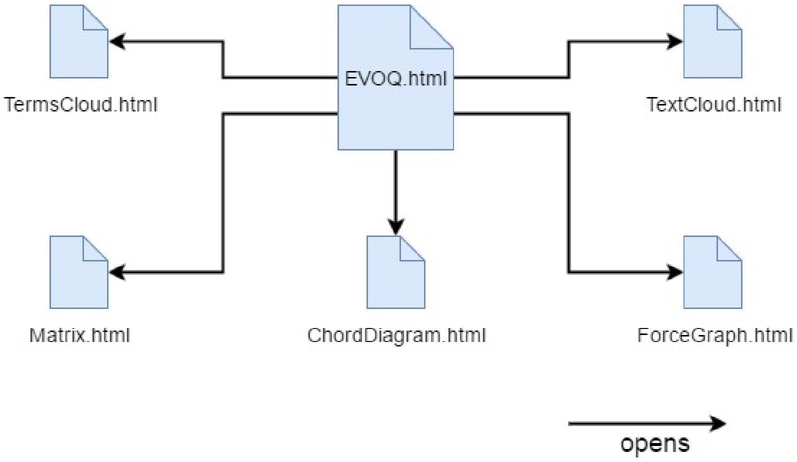


Figure 3.6: HTML files in STAVIZ

This architecture has advantages and drawbacks.
Advantages

- Adding a new visualization is straightforward : it can be achieved by creating a new HTML file and creating an HTML element to access it (for instance, a button) in the EVOQ.html file

- When a visualization needs to be modified, there is only one file to edit
- Modifying the code of a visualization will have no side effect on the main page nor on the other visualizations

Drawbacks

- Linking-and-brushing between the visualizations is not straightforward since they are on different pages

Javascript Files The Javascript files comprise Javascript functions that are called by the visualization pages and by the main page. They are divided into two categories.

Firstly, some Javascript files are written as a constructor function. This is how Javascript emulates object classes. Each of these files represent a concept of structural analysis (term, terms dictionary, relationship, relationships dictionary) and provide functions to interact with them such as getters, setters, and display functions. Figure 3.7 is a simplified class diagram showing the relationships between these concepts, and subsequently between the Javascript files.

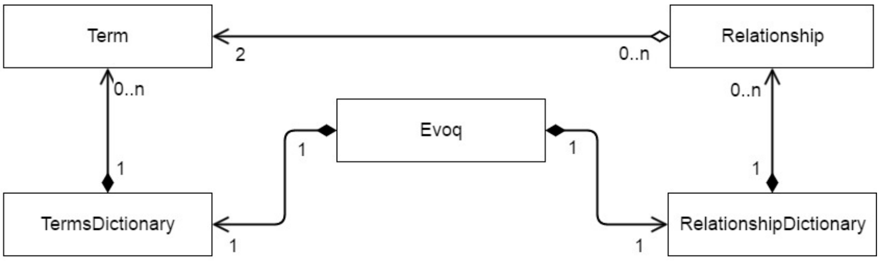


Figure 3.7: Simplified class diagram of the structural analysis concepts in STAVIZ. *Evoq* represents an analysis workspace, that is, the composition of a text, a terms dictionary, and a relationships dictionary.

Secondly, other Javascript files are written as a set of functions that address a cross-cutting concern in STAVIZ. For instance,

- EVOQparsing.js handles the parsing of JSON into concepts of structural analysis
- EVOQlang handles the language-specific concerns. More precisely, this consists of word stemming functions and stopwords lists
- EVOQtextProcessing provides functions for processing textual data (stopwords removal, special characters removal, bag of words extraction)

These functions are called from the HTML files and from the Javascript object files. They are therefore separated from them for reusability and maintainability purposes.

CSS Files The CSS files are responsible for the aesthetics in STAVIZ. CSS code can be written into a dedicated CSS file or inside *style* tags into a HTML file. When the CSS was specific to a HTML file, it was written into it. Otherwise it was written in a CSS file to allow other HTML files to reuse it.

External Libraries The external libraries used in STAVIZ are listed below :

- D3.js ((Bostock *et al.*, 2011))
  Use : generating the visualizations
  Website : `https://d3js.org/`
- jQuery
  Use : easing Javascript development (HTML elements manipulation, event handling, API querying, ...)
  Website : `https://jquery.com/`
- lemmatizer
  Use : English stemming
  Website : `https://github.com/takafumir/javascript-lemmatizer`
- snowball
  Use : French stemming
  Website : `http://snowball.tartarus.org/algorithms/french/stemmer.html`
- guessLanguage
  Use : guessing the language of a text because stemming is language-dependant
  Website : `https://github.com/richtr/guessLanguage.js/`

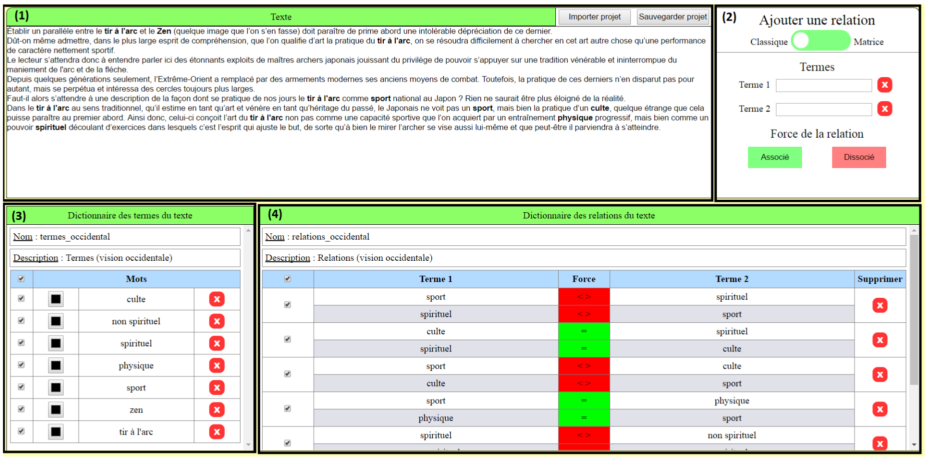
Credit is also given to Michael Bostock[3] [4], Eric Coopey[5], and AndrewRP (pseudonym)[6] for their examples of D3.js use.

3.6 Functionality

3.6.1 STAVIZ Main Page

As an attempt to implement the structural analysis technique, STAVIZ allows typing, pasting and importing (.txt format) a text. It is then possible to define a relationship between two terms of the user's choosing. The encoded relationships can be either disjunctions or conjunctions. The user can see two table visualizations : one for the relationships that are defined for the text and the other one for the terms in the relationships. They are respectively named *relationships dictionary* and *terms dictionary*.
These features are presented in the main page (Figure 3.8). The main page is the page that is shown to the users when they launch STAVIZ.

[3] https://bl.ocks.org/mbostock/3750558 (accessed December 7, 2016)
[4] https://bl.ocks.org/mbostock/4062045 (accessed October 6, 2016)
[5] http://bl.ocks.org/ericcoopey/6382449 (accessed October 6, 2016)
[6] http://bl.ocks.org/AndrewRP/7468330 (accessed October 17, 2016)



(1) Texte

Importer projet | Sauvegarder projet

Établir un parallèle entre le **tir à l'arc** et le **Zen** (quelque image que l'on s'en fasse) doit paraître de prime abord une intolérable dépréciation de ce dernier.
Dût-on même admettre, dans le plus large esprit de compréhension, que l'on qualifie d'art la pratique du **tir à l'arc**, on se résoudra difficilement à chercher en cet art autre chose qu'une performance de caractère nettement sportif.
Le lecteur s'attendra donc à entendre parler ici des étonnants exploits de maîtres archers japonais jouissant du privilège de pouvoir s'appuyer sur une tradition vénérable et ininterrompue du maniement de l'arc et de la flèche.
Depuis quelques générations seulement, l'Extrême-Orient a remplacé par des armements modernes ses anciens moyens de combat. Toutefois, la pratique de ces derniers n'en disparut pas pour autant, mais se perpétua et intéressa des cercles toujours plus larges.
Faut-il alors s'attendre à une description de la façon dont se pratique de nos jours le **tir à l'arc** comme **sport** national au Japon ? Rien ne saurait être plus éloigné de la réalité.
Dans le **tir à l'arc** au sens traditionnel, qu'il estime en tant qu'art et vénère en tant qu'héritage du passé, le Japonais ne voit pas un **sport**, mais bien la pratique d'un **culte**, quelque étrange que cela puisse paraître au premier abord. Ainsi donc, celui-ci conçoit l'art du **tir à l'arc** non pas comme une capacité sportive que l'on acquiert par un entraînement **physique** progressif, mais bien comme un pouvoir **spirituel** découlant d'exercices dans lesquels c'est l'esprit qui ajuste le but, de sorte qu'à bien le mirer l'archer se vise aussi lui-même et que peut-être il parviendra à s'atteindre.

(2) Ajouter une relation

Classique ○ Matrice

Termes

Terme 1 ☒

Terme 2 ☒

Force de la relation

Associé | Dissocié

(3) Dictionnaire des termes du texte

Nom : termes_occidental

Description : Termes (vision occidentale)

☑	Mots		
☑	■	culte	☒
☑	■	non spirituel	☒
☑	■	spirituel	☒
☑	■	physique	☒
☑	■	sport	☒
☑	■	zen	☒
☑	■	tir à l'arc	☒

(4) Dictionnaire des relations du texte

Nom : relations_occidental

Description : Relations (vision occidentale)

☑	Terme 1	Force	Terme 2	Supprimer
☑	sport	< >	spirituel	☒
	spirituel	< >	sport	
☑	culte	=	spirituel	☒
	spirituel	=	culte	
☑	sport	< >	culte	☒
	culte	< >	sport	
☑	sport	=	physique	☒
	physique	=	sport	
☑	spirituel	< >	non spirituel	☒

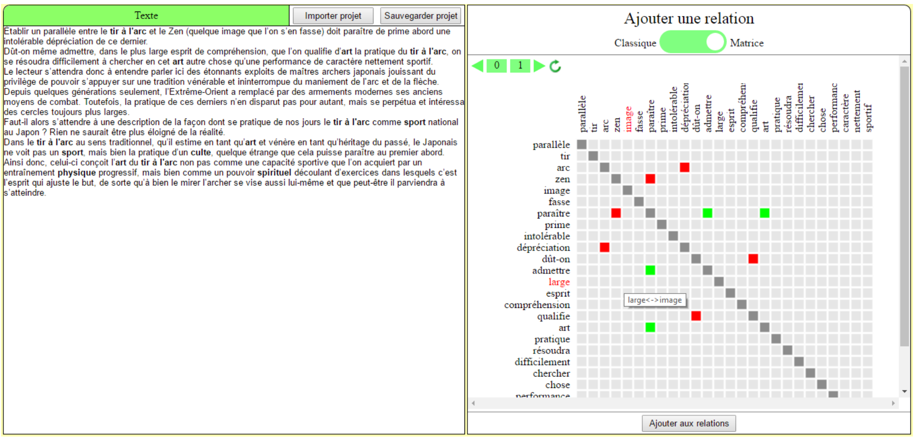
Figure 3.8: STAVIZ main page

The main page comprises four distinct parts, one for each aforementioned feature:

- Text (1)
- Relationships encoding (2)
- Terms dictionary (3)
- Relationships dictionary (4)

The main page also offers an alternative relationships encoding module. It allows adding relationships faster by selecting them on an adjacency matrix of terms rather than selecting the two terms manually. The user can access the matrix relationships encoding by clicking on the switch at the top of the relationships encoding box.
This also hides the terms and the relationships dictionary in order to allocate enough space to the matrix. However, any relationship in the dictionary is displayed in the matrix when its two terms are also displayed.

Figure 3.9 shows the main page with the matrix encoding settings. The user can return to the standard encoding by hitting the switch again.



Établir un parallèle entre le **tir à l'arc** et le Zen (quelque image que l'on s'en fasse) doit paraître de prime abord une intolérable dépréciation de ce dernier.
Dût-on même admettre, dans le plus large esprit de compréhension, que l'on qualifie d'**art** la pratique du **tir à l'arc**, on se résoudra difficilement à chercher en cet **art** autre chose qu'une performance de caractère nettement sportif.
Le lecteur s'attendra donc à entendre parler ici des étonnants exploits de maîtres archers japonais jouissant du privilège de pouvoir s'appuyer sur une tradition vénérable et ininterrompue du maniement de l'arc et de la flèche.
Depuis quelques générations seulement, l'Extrême-Orient a remplacé par des armements modernes ses anciens moyens de combat. Toutefois, la pratique de ces derniers n'en disparut pas pour autant, mais se perpétua et intéressa des cercles toujours plus larges.
Faut-il alors s'attendre à une description de la façon dont se pratique de nos jours le **tir à l'arc** comme **sport** national au Japon ? Rien ne saurait être plus éloigné de la réalité.
Dans le **tir à l'arc** au sens traditionnel, qu'il estime en tant qu'**art** et vénère en tant qu'héritage du passé, le Japonais ne voit pas un **sport**, mais bien la pratique d'un **culte**, quelque étrange que cela puisse paraître au premier abord.
Ainsi donc, celui-ci conçoit l'**art** du **tir à l'arc** non pas comme une capacité sportive que l'on acquiert par un entraînement **physique** progressif, mais bien comme un pouvoir **spirituel** découlant d'exercices dans lesquels c'est l'esprit qui ajuste le but, de sorte qu'à bien le mirer l'archer se vise aussi lui-même et que peut-être il parviendra à s'atteindre.

Figure 3.9: STAVIZ main page with the matrix encoding

The matrix comprises the terms for two consecutive paragraphs (p, p+1) of the text, or the terms of one paragraph if the text has only one. The user can generate the adjacency matrix for paragraphs (p-1, p) or (p+1, p+2) with the green triangles above the matrix. The reason why the matrix works this way is because a structural analyst works this way too : they typically analyze a text as a flow of sentences and it is convenient for them to have a view of only two paragraphs at once.

STAVIZ offers various functionality for the text, the terms dictionary, and the relationships dictionary. They can be accessed from the main page by hovering the header of these parts. Figure 3.10 (resp. 3.11, 3.12) shows the functionality available for the text (resp. terms dictionary, relationships dictionary).

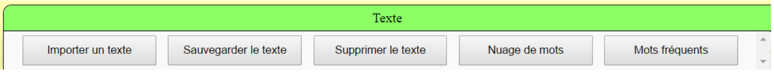


Figure 3.10: Text functionality

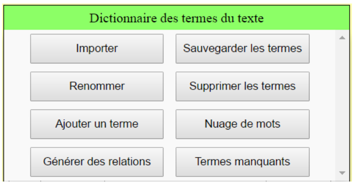


Figure 3.11: Terms dictionary functionality

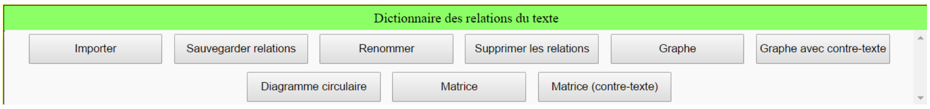


Figure 3.12: Relationships dictionary functionality

Every feature has a usefulness, but some are of much more interest in the context of this thesis, namely those which consist of a visualization. These are :

- Relationships suggestion
- Word cloud for the text
- Word cloud for the terms dictionary
- Node-link diagram for the relationships dictionary
- Chord diagram for the relationships dictionary
- Matrix for the relationships dictionary

They are detailed in the following subsections.

3.6.2 Relationships Suggestion

The relationships suggestion feature aims at helping the structural analyst in the relationships discovery process. It can help them to discover a relationship that doesn't appear clearly in the text (implicit) or an explicit one that was missed in the process.
There are three algorithms implemented in STAVIZ that generate relationships. They are structured as follows :

```
Relationships-based
    > Implicit relationships suggestion
Terms-based
    > Wikipedia-based suggestion
    > Text-based suggestion
```

Implicit Relationships Suggestion The implicit relationships suggestion uses a rule from the structural analysis technique to infer relationships from the relationships dictionary.

This relationship inference rule is applied as follows :

```
Let A, B, C, and D be terms. (1)
Let A-B and C-D be disjunction relationships. (2)
Let A-C be a conjunction relationship. (3)

Given (1), (2), and (3), B-D is an implicit conjunction relationship.
```

The rule is illustrated in Figure 3.13.

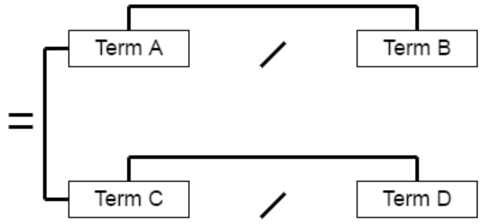


Figure 3.13: Implicit relationships inference rule

It is worth noting that the relationships inference is not entirely reliable as false positives and false negatives may occur. Since the structural analysis is used, the analyst should always have the final say. The analyst should review the results returned by the relationships suggestion module and remove the relationships they deem irrelevant.

Wikipedia-Based and Text-Based Suggestion The terms-based relationships suggestion infers relationships by representing each term by a bag of words. In turn, a similarity index is computed for each pair of bags of words. If the similarity index is high enough, a relationship is inferred. This process is illustrated in Figure 3.14.

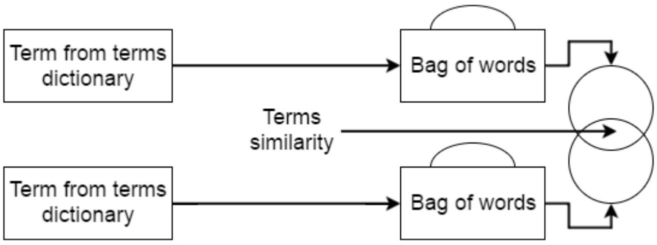


Figure 3.14: Process for terms-based relationships suggestion

The two implementations of the terms-based approach in STAVIZ are the Wikipedia-based suggestion and the text-based suggestion. The difference between them is how the bag of words is generated from a term.

- Wikipedia-based : the bag of words is generated with the first sentence of the Wikipedia page. If no unique Wikipedia page could be found for the term, it will not be considered for the relationships suggestion. This approach draws on (Panchenko *et al.*, 2012).
  The following is a simplified code snippet from STAVIZ that handles queries to the Wikipedia API.

```
function retrieveData(word){
    var URLbegin =
        "https://en.wikipedia.org/w/api.php?action=query&format=
     json&prop=extracts|categories&titles=";
    var URLend = "&exintro&utf8&callback=?";
    URL = URLbegin + word.replace(/ /g, "+") + URLend; // Build
        URL with the word we want to look up a page for
    $.getJSON(URL, function(data){
        var queryresult = data.query.pages; // Get the
            Wikipedia pages
        // Check here if pages were returned
        // If pages are returned, check their category here to
            detect disambiguation pages
    });
}
```

- Text-based : the bag of words is generated with all the sentences of the analyzed text in which the term appears

In order to provide the analyst with more accurate results, the generated bags of words are cleaned before similarity is computed. The function for this use performs two cleanings. Firstly, it replaces any number or any special character by the empty string. Secondly, it removes stopwords and duplicates.

After this cleaning step, the similarity index can be computed for each pair of bag of words. For each term X, the three other terms that have the highest similarity index (T1, T2, and T3) are selected. The relationships X-T1, X-T2, and X-T3 are suggested to the user. There may be less than three relationships suggested since the terms for which the similarity index is 0 are not considered.

The type of the relationship is not specified. It is up to the user to decide if two terms form a conjunction or a disjunction.

Finally, the suggested relationships are displayed on an adjacency matrix of terms. Figure 3.15 shows the relationships suggested for the terms of the node-link diagram in Figure 3.19. An orange-colored (resp. black-framed) square means that a relationship was suggested by the text-based (resp. Wikipedia-based) algorithm. An asterisk written next to a term means that no unique Wikipedia page could be found for this term.

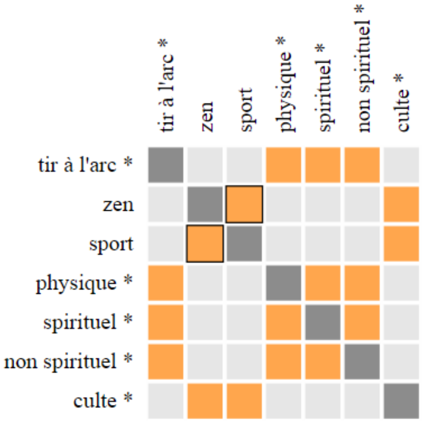


Figure 3.15: Relationships suggested for the terms of the node-link diagram in Figure 3.19

The matrix on Figure 3.15 also illustrates the limitations of the implemented terms-based approaches.

Limitations of the Wikipedia-based implementation :

- Context is important in structural analysis and the Wikipedia-based does not take the context into account
- A term is not represented by many words, thus similarity indexes of 0 occur repeatedly and few relationships are suggested

- A term for which no unique Wikipedia page could be found is not considered, interesting relationships could consequently be missed

- Not finding a unique Wikipedia page occurs often for groups of words and for non-English words. This is because here are much less Wikipedia pages for other languages than English

- The user should always review the suggested relationships to remove the false positives

- The user cannot entirely rely on the Wikipedia-based suggestion. There will certainly be false negatives

Limitations of the text-based implementation :

- When the text is short, a term is not represented by many words. Thus, similarity indexes of 0 occur repeatedly and few relationships are suggested

- The user should always review the suggested relationships to remove the false positives

- The user cannot entirely rely on the text-based suggestion. There will certainly be false negatives

Figure 3.16 shows the relationships suggested for 13 terms from Martin Luther King's speech *I have a dream*. It illustrates that the text-based implementation can prove insightful when the text is large enough. However, the Wikipedia-based implementation shows again barren results for this example.

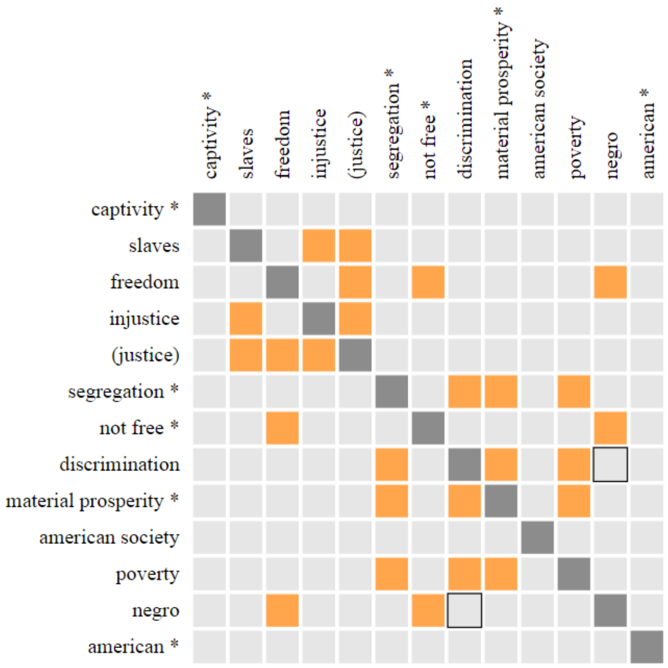


Figure 3.16: Relationships suggested for 13 terms from Martin Luther King's speech *I have a dream*

3.6.3 Word Cloud

Word Cloud for the Text The word cloud for the text takes the 30 most frequent terms (either words or groups of words) of the text and displays them on a word cloud on a new page.

As in a traditional word cloud, a term appears bigger in size if it is more frequent in the text. Since the word cloud in STAVIZ does not handle collisions, the opacity of more frequent terms is lowered. This allows seeing less frequent words when they collide with frequent ones. Figure 3.17 shows the word cloud generated for Martin Luther King's well-known speech *I have a dream.* It illustrates the issue of colliding words and how the opacity management solves it.

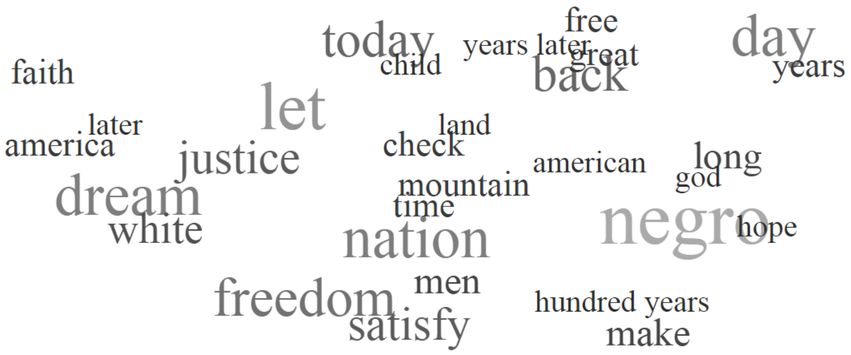


Figure 3.17: Word cloud generated for Martin Luther King's speech *I have a dream*

In order to make the word cloud more helpful, interactions possibilities were added :

- Words can be moved by a drag and drop
- The color of the words can be changed
- Words can be hidden and brought back

With these possibilities, the users can go fairly far in the customization of the cloud and hopefully make a word cloud that is genuinely their own. The color change and word hiding can be accessed by right-clicking on a word. This action displays the menu shown in Figure 3.18.

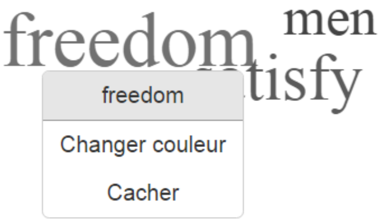


Figure 3.18: Interaction menu for the word *freedom*

The word cloud does not display any relationship between terms and may seem irrelevant to structural analysis in this respect. However, it gives a fairly simple yet accurate overview of a text and could prove convenient if the analyst wishes for an overview before they proceed with the structural analysis. It gives a glimpse of the most important words for the author of the text. It is the user's choice whether to use it or not, but it can provide a good starting point in this respect.

Word Cloud for the Terms Dictionary The word cloud for the terms dictionary is the same as the word cloud for the text. It offers the same interaction possibilities. Its goal is to provide an alternative view of the terms dictionary that is more convenient than a scrollable table of terms.

The only change between the two word clouds is the origin of the set of words. The word cloud for the terms dictionary generates a cloud from the terms in the dictionary. It thus allows generating a cloud with terms of the user's choosing.

3.6.4 Node-link Diagram

The node-link diagram represents the relationships set as a bi-dimensional graph. The terms (resp. relationships) relate to the nodes (resp. the links between two given nodes).

A node is represented as a colored disk with the corresponding term written above it and a link is represented as a colored line drawn from a node to another. The color of the line depends on the type of the relationship it represents : green for a conjunction and red for a disjunction. These colors were chosen because they remind respectively of association and opposition. On the other hand, the color of a node is of the user's choosing.
Figure 3.19 shows an example of a node-link diagram generated by the following relationships set :

```
zen <disjunction> tir à l'arc
sport <conjunction> tir à l'arc
sport <conjunction> physique
culte <disjunction> sport
sport <disjunction> spirituel
culte <conjunction> spirituel
spirituel <conjunction> non spirituel
```

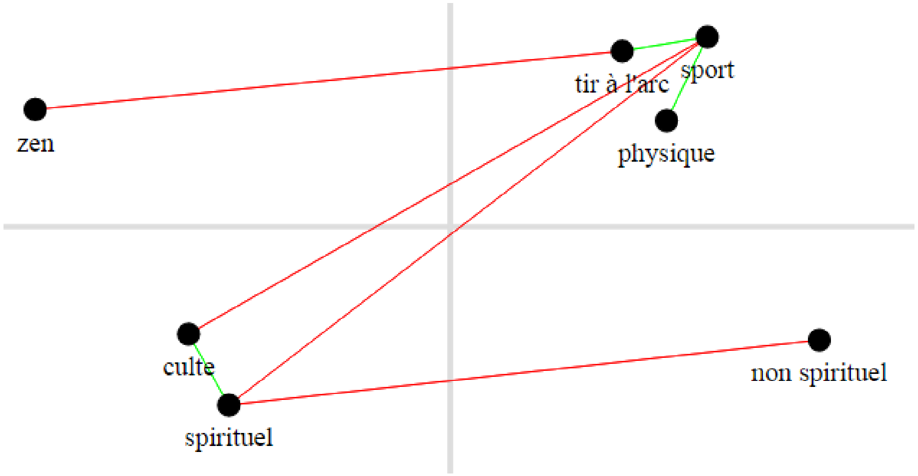


Figure 3.19: Example of node-link diagram

Since the analyst's interpretation has a critical role in structural analysis, the node-link visualization must provide ways to extract knowledge from itself. This was implemented by several interaction possibilities offered to the user.
More precisely, these are :

- Hide all the links to better observe the node layout when there is link clutter. Figure 3.20 shows the node-link diagram in Figure 3.19 with hidden links.

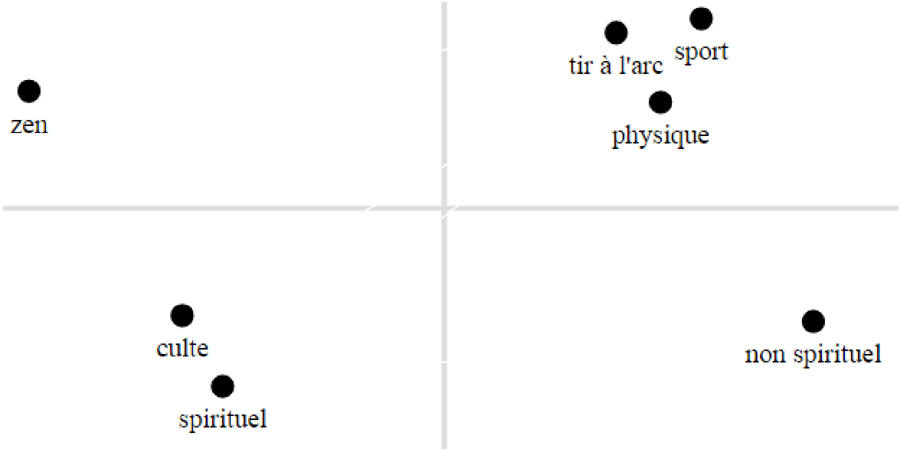


Figure 3.20: Node-link diagram in Figure 3.19 with hidden links

- Drag and drop a node
- Change the color of a node
- Fix or unfix the position of a node
- Add or remove links

- Focus one or more node (this reduces the opacity of any node or link which is not connected to a focused node). Figure 3.21 shows the node-link diagram in Figure 3.19 with the term *spirituel* on focus.

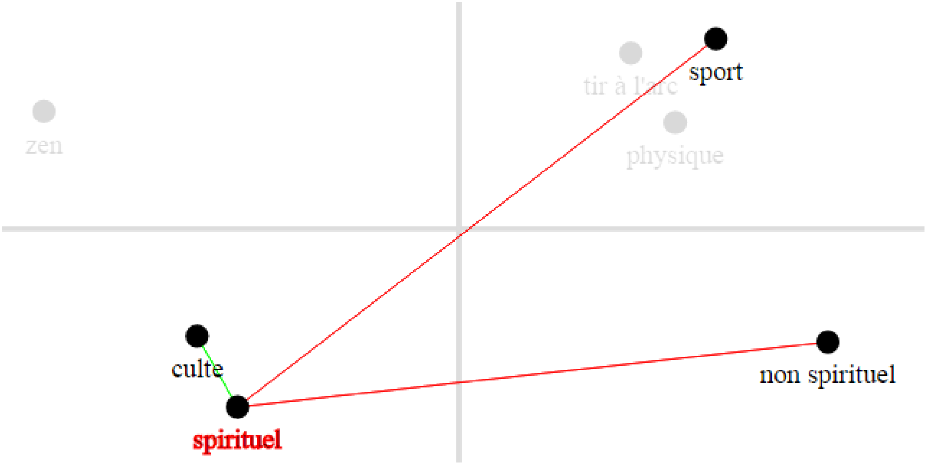


Figure 3.21: Node-link diagram in Figure 3.19 with the term *spirituel* on focus

The interaction functionality for a node (resp. link) can be accessed by a right click on it as illustrated by Figure 3.22 (resp. Figure 3.23).

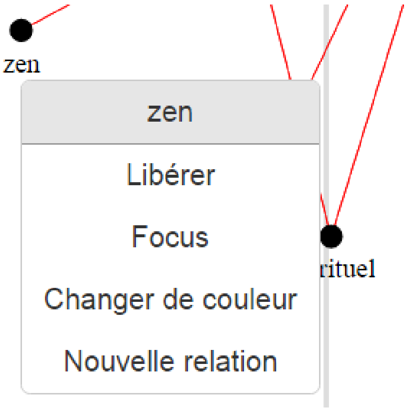


Figure 3.22: Interaction functionality for a node

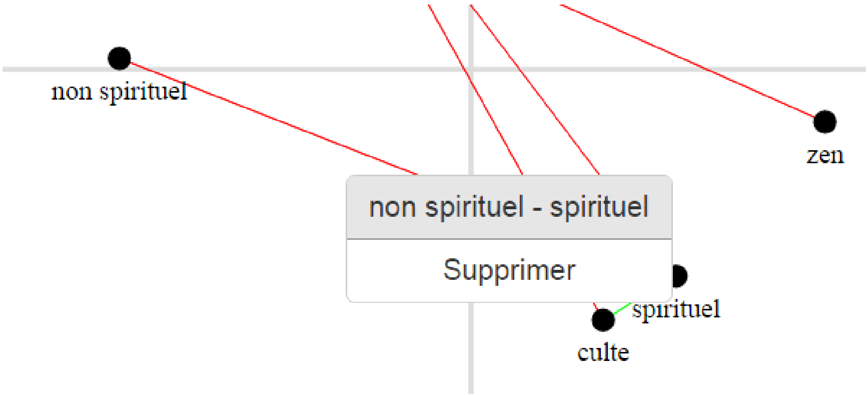


Figure 3.23: Interaction functionality for a link

The drag and drop of a node is the most important interaction feature. When a user drags a node, they can observe the effect it has on the semantic field represented by the diagram. This is precisely what an analyst who performs structural analysis is looking for because it is the type of knowledge they need and that is difficult to get without a visualization. This is what makes the node-link diagram the visualization that brings the most added-value to the analyst.

The focus, links hiding, and the drag and drop interactions allow the user to reduce the number of visible edge crossings in the graph. In this regard, they improve the legibility of the diagram. The number of edge crossings is indeed the most penalizing issue for legibility in a node-link diagram according to (Purchase, 2000).

The layout of the diagram is generated by a call to the function

```
d3.layout.force()
```

from he d3.js librairy. The complete code for the layout is the following.

```
var force = d3.layout.force()
.size([width, height]) // Size of the diagram (in pixels)
.charge(-400)  // Force-layout (forces the nodes to repel each other
    during the layout to prevent node overlapping)
.linkDistance(function(link){
  return link.value*30 + 40; // Forces the distance between the
      connected nodes (small distance (40 pixels) for conjunction,
      bigger distance for conjunction)
})
.on("tick", tick); // Adaptative behavior of the diagram when a node
    is moved
```

3.6.5 Chord Diagram

The chord diagram is an alternative representation to the node-link diagram for representing a relationships set. The arcs represent the terms and the chords connecting these arcs represent the relationships.

The thickness of a chord depends on the type of the relationship it represents. The disjunction relationships are thicker than the conjunctions because they are more important for a structural analyst. As in the node-link diagram, the color of the arcs is of the user's choosing. The color of a chord is the average of the color of the two arcs it connects.
Figure 3.24 and 3.25 respectively show a node-link diagram with colored nodes and the chord diagram that represents the same relationships set. The diagram on Figure 3.24 represents the relationships in Figure 3.19 and the implicit relationships suggested from these.

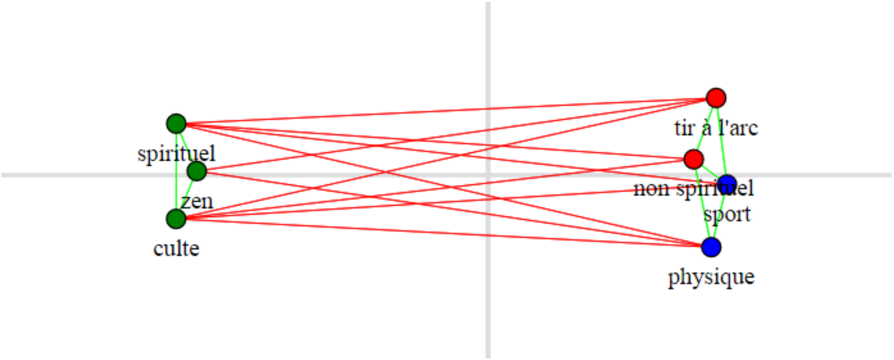


Figure 3.24: Node-link diagram representing the relationships in Figure 3.19 and the implicit relationships suggested from these

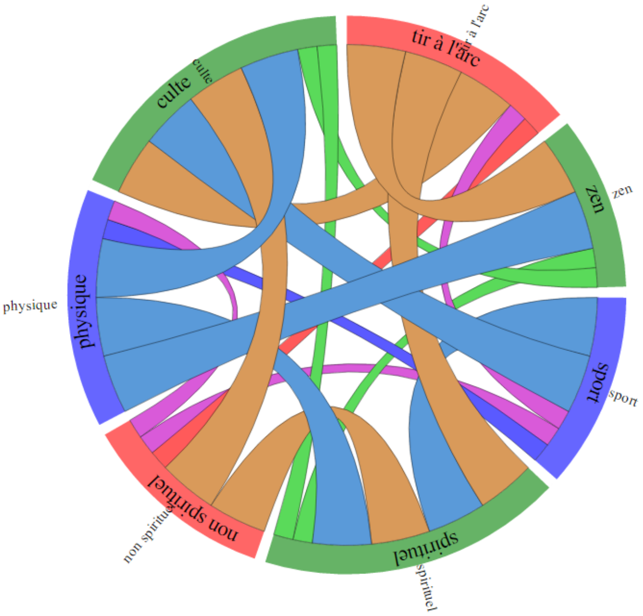


Figure 3.25: Chord diagram representing the relationships in Figure 3.24

The chord diagram was originally implemented to solve the link clutter issue that emerges when there are too many relationships (see Figure 2.30). It was later replaced by the focus interaction in the node-link diagram.

It was left in STAVIZ because some users may find it convenient to use when the node-link diagram resembles a ball of wool, which happens often when the implicit relationships are added to the diagram. However, its interaction possibilities are more limited than the node-link diagram's. The user can :

- Change the color of an arc (this subsequently changes the color of the connected chords)
- Focus one or more arc (this reduces the opacity of any chord which is not connected to a focused arc). Figure 3.26 shows the chord diagram in Figure 3.25 with the term *sport* on focus.

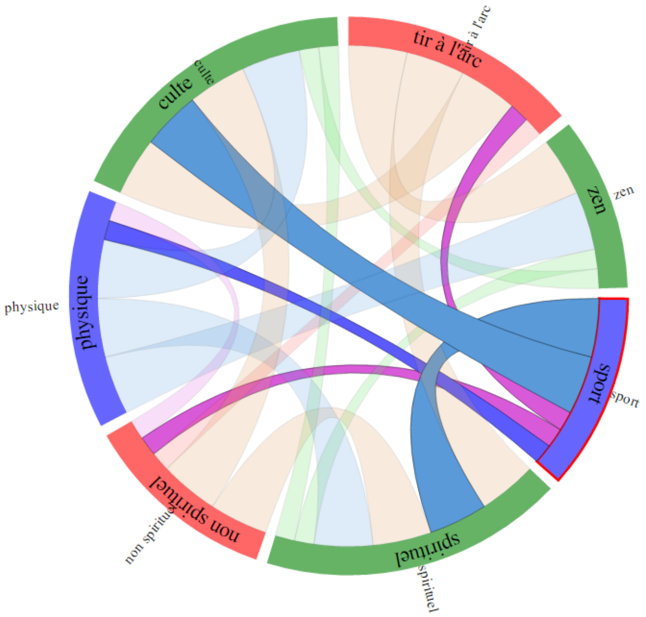


Figure 3.26: Chord diagram in Figure 3.25 with the term *sport* on focus

The interaction functionality can be accessed by a right-click on an arc, as illustrated in Figure 3.27.

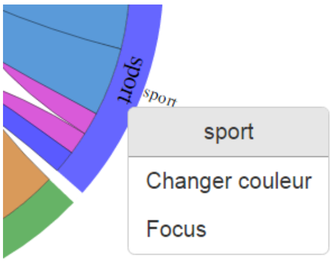
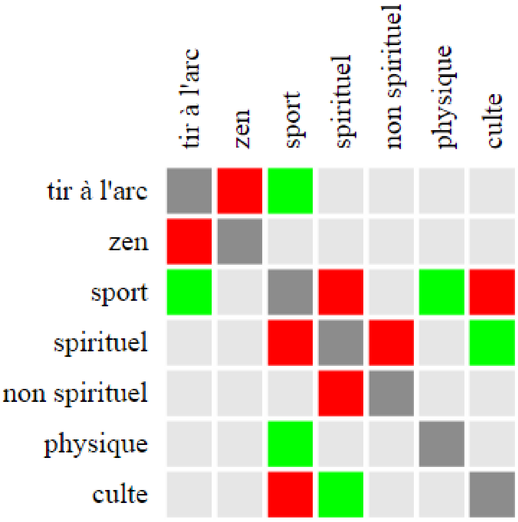


Figure 3.27: Interaction possibilities for the chord diagram

3.6.6 Matrix

The matrix visualization represents a set of relationships as an adjacency matrix of terms. It is symmetric since relationships are bidirectional.

The matrix is presented in the same way as the matrix for relationships encoding in Figure 3.9.
Figure 3.28 shows the relationships from Figure 3.19 represented in the matrix.

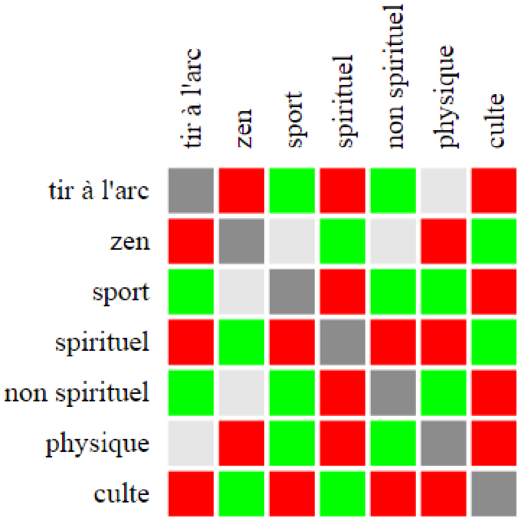


	tir à l'arc	zen	sport	spirituel	non spirituel	physique	culte
tir à l'arc	◼	🟥	🟩	⬜	⬜	⬜	⬜
zen	🟥	◼	⬜	⬜	⬜	⬜	⬜
sport	🟩	⬜	◼	🟥	⬜	🟩	🟥
spirituel	⬜	⬜	🟥	◼	🟥	⬜	🟩
non spirituel	⬜	⬜	⬜	🟥	◼	⬜	⬜
physique	⬜	⬜	🟩	⬜	⬜	◼	⬜
culte	⬜	⬜	🟥	🟩	⬜	⬜	◼

Figure 3.28: Matrix representation of the relationships in Figure 3.19

As shown by Ghoniem et al. in (Ghoniem *et al.*, 2005), the matrix visualization proves to be more efficient than the node-link diagram for larger relationships sets. The main reason for this is the matrix's insensitiveness to the relationships set size.
This is especially useful when the implicit relationships are added to the set. As mentioned earlier, they often cause the node-link diagram to resemble a ball of wool.

Figure 3.29 shows the matrix visualization of the relationships in Figure 3.24. A comparaison between the node-link diagrams on Figure 3.19 and 3.24 on the one hand and between the matrices on Figure 3.28 and 3.29 on the other hand illustrate that the matrix handles implicit relationships better than the node-link diagram.



	tir à l'arc	zen	sport	spirituel	non spirituel	physique	culte
tir à l'arc	⬛	🟥	🟩	🟥	🟩	⬜	🟥
zen	🟥	⬛	⬜	🟩	⬜	🟥	🟩
sport	🟩	⬜	⬛	🟥	🟩	🟩	🟥
spirituel	🟥	🟩	🟥	⬛	🟥	🟥	🟩
non spirituel	🟩	⬜	🟩	🟥	⬛	🟩	🟥
physique	⬜	🟥	🟩	🟥	🟩	⬛	🟥
culte	🟥	🟩	🟥	🟩	🟥	🟥	⬛

Figure 3.29: Matrix representation of the relationships in Figure 3.24

However, the matrix visualization has a major drawback compared to the node-link diagram : it lacks interaction possibilities. It is difficult to detect groups of terms or to see how disrupting of an influence a term will have on the semantic field. This is highly penalizing in structural analysis.

3.7 Linking and Brushing

Linking and brushing is defined by Keim in (Keim, 2002) as *"the idea to combine different visualization methods to overcome the shortcomings of single techniques"*. Linking and brushing also implies that when the user makes changes to one visualization, the others should be consequently impacted. Keim also argues that visualizations bound with linking and brushing are more informative than the same visualizations considered independently.

Although linking and brushing was not fully implemented, STAVIZ offers features in this regard. Table 3.1 summarizes which interactions are reflected, as well as the source and the target visualizations of the repercussions.

Table 3.1: Linking and brushing in STAVIZ

Interaction	Source visualizations	Target visualizations
Change color	Any of the following : main page, word cloud, node-link diagram, chord diagram	All the others (except the matrix)
Add a relationship	Main page (resp. node-link diagram)	Node-link diagram (resp. main page)
Delete a relationship	Main page (resp. node-link diagram)	Node-link diagram (resp. main page)
Edit a term	Main page	All the others (except the matrix)
Hide a term	Main page (resp. word cloud)	Word cloud (resp. main page)
Show a hidden term	Main page (resp. word cloud)	Word cloud (resp. main page)

Further work could focus on improving the linking and brushing feature. For instance, including the matrix visualization and the focus interaction would be relevant improvements.

3.8 Evaluation With Users

An evaluation session with users was organized on November 28, 2016, to assess the usability and the usefulness of STAVIZ.

3.8.1 Participants

The participants of the session were a management masters student from the University of Namur and a researcher from the Law Faculty of the university of

Namur. They were asked to bring a text of their choosing to analyze.

3.8.2 Methodology

The session began with a brief introduction of the fundamental concepts of structural analysis. Afterwards, STAVIZ was installed on the participants' laptop. They were then faced with STAVIZ and were asked to proceed with structural analysis without explanation on how the software works. The evaluation technique used for this step was the think-aloud technique. It consists of asking the participants to express aloud their actions and how they feel about their use of STAVIZ.

When the participants faced difficulties they were encouraged to try to solve the problem for one or two minutes before receiving help. This allowed measuring to what extent the difficulty is complex to deal with as well as the time a user is ready to spend to attempt to solve the problem before giving up their task or resorting to external help.

The advantage of this evaluation technique is that it allows understanding how the users would actually use STAVIZ and deal with encountered difficulties. It also allows gathering insightful feedback on the design and the functionality. For example, the users could express their interest for the features they consider the most relevant, and they could express their confusion about unclear elements of design.

3.8.3 Results

The evaluation lasted two hours. This time was not enough for the participants to explore STAVIZ in its entirety because the design issues of the main page constituted a heavy obstacle to the fulfillment of their task. Overall, the participants explored the main page and briefly the word cloud.

Since the visualizations could not be covered by the user evaluation, the problems reported regarded mainly design issues. Examples are provided below :

- The option menus are difficult to access
- The label of some buttons is unclear
- When a *save* button is clicked, it is unclear what is actually saved

Solving the majority of the design issues took approximately one week. The insightful feedback received from the participants allowed for a significantly clearer design of the main page.

The functionality uncovered in the user evaluation were explained to the participants subsequently to the session. Despite the aforementioned design issues, they expressed their enthusiasm towards the possibilities offered by STAVIZ and agreed to take part to a subsequent evaluation.

3.9 Discussion

The current version of STAVIZ was presented to the members of the EFFaTA-MeM projects in December 2016. They gave a positive feedback to STAVIZ, noting the following contributions :

- STAVIZ was developed using more recent and more suitable technologies than the previous EVOQ
- New visualization techniques were explored, although the most relevant one remains the node-link diagram
- The relationships encoding is much faster with the adjacency matrix
- Despite having limitations, a relationships suggestion module was developed following several different approaches

However, they noted that there are possibilities of improvements and future work.

Furthermore, the development methodology has shortcomings. As a first experience with visualizations development, STAVIZ was not developed following the prescribed good practices. A non-exhaustive list of methodology faults is provided below.

- The understanding of the users and their tasks was not extensive enough in the beginning of the development
- The first evaluation with users took place three weeks before the end of the implementation. One week was needed to solve the main design issues. As a result, not all parts of STAVIZ underwent the evaluation
- The design should have been discussed with users through paper sketching early in the implementation phase

A compliance to the listed good practices and to other guidelines such as Tufte's mentioned in (Culy et Lyding, 2009) would undoubtedly have led to significantly better results.

Chapter 4

Proposing an Evaluation Grid

Section 3.9 shows that the proposed STAVIZ is not yet perfectly suited for helping a user to perform structural analysis. This chapter aims at proposing an evaluation grid which contains the criteria to decide whether a visualization technique or tool is suited for this task. It is organized as follows.

First, fundamental concepts of structural analysis are presented. Then, these are used to define a list of as objective as possible criteria that compose the evaluation grid. In turn, STAVIZ and the techniques mentioned in the state of the art will be evaluated with the proposed grid. Afterwards, improvements of STAVIZ are proposed with the purpose to match thoroughly the grid. Finally, a paper prototype is built to illustrate how the proposed improvements could be implemented.

4.1 Fundamental Concepts

The structural approach is interested in understanding the discourse of a speaker at the perception level, that is, understanding the system of representations inside which a speaker makes their discourse (Wallemacq et Jacques, 2001). This system of representations is a set of associations and oppositions between terms called the *semantic field*. The meaning of terms (denotation) is irrelevant in this approach. A term is defined by the other terms with which it has an association or an opposition relationship (connotation). The analyst must uncover these relationships in the text to reconstitute the set of relationships defined by the speaker.

Another key concept of the approach is the idea that the page is never blank. It means that when someone reads or hears the discourse of a speaker, they won't consider only the relationships set built by the speaker. It is because the speaker expresses himself in a language that already carries a set a relationships. The following passage from Wallemacq and Jacques in (Wallemacq et Jacques, 2001) further illustrates this idea :

"All speech is bound to a language which carries within itself its own vision of the world. And this vision of the world, again with the situation in ethnomethodology, is not universal but belongs to the user of a particular language."

This implies that the speaker doesn't have full control of the meaning of the text he produces. The set of relationships of the speaker exists only on top of the language's set. The latter constantly threatens the former because the meaning of the discourse is decided by both at the end. This is called the *"deferred meaning"*, or the *"sense on the rebound"*.

Wallemacq and Jacques provide a simple yet illustrative example to embody this idea in (Wallemacq et Jacques, 2001).

"When at our university we speak of "management", management is in fact opposed to "economics", and whether we like it or not, we have to deal with a semantic field which assigns management to the realm of the "material", of the "practical", while economics belongs on the level of the "theoretical", of the "pure", the level of "fundamental research". As if by symmetry, management becomes the domain of the "applied"n the "non-pure". Also reciprocally, since "management" is "applied", it belongs to the domain of what is "useful", the realm of things done, no longer merely talked about."

Figure 4.1 shows the node-link diagram generated in STAVIZ for this example. The sets of relationships are defined as follows :

```
Legend : <> Opposition relationship
          = Association relationship

Speaker's relationships set
---------------------------------------------
management <> economics

Language's relationships set
---------------------------------------------
management = material
management = practical
economics = theoretical
economics = pure
economics = fundamental research
pure <> non-pure
theoretical <> applied
```

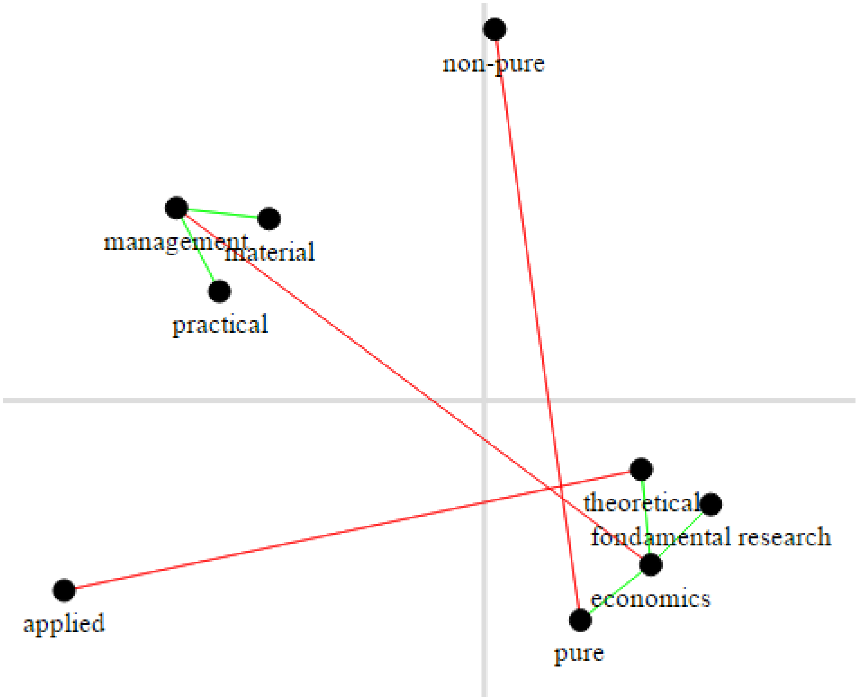


Figure 4.1: Node-link diagram generated in STAVIZ for the management-economics example

When the implicit relationships are added in STAVIZ, the diagram in Figure 4.2 shows the relationships found by symmetry in the example.

```
Implicit relationships
-----------------------------------------------
management = applied
management = non-pure
```

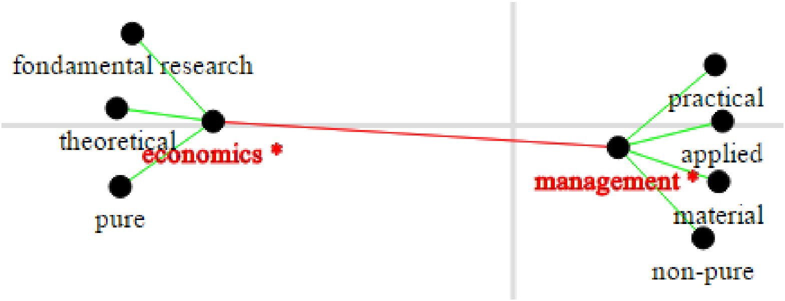


Figure 4.2: Node-link diagram generated in STAVIZ for the management-economics example (implicit relationships included, focus on *management* and *economics* for legibility)

The discourse of the author consists of only one opposition relationship between *management* and *economics*. After adding the relationships from the language,

many more relationships are to be considered. This example demonstrates how the language can threaten the discourse of the speaker. From *Management is opposed to economics*, we have *Management is not-pure*, which are in no way the words that the speaker said, but are nonetheless the words conveyed by his discourse.
As de Saint-George writes in her summary about EVOQ (Saint-Georges, 2004), the representation system of a speaker can be fully understood only when the the set relationships of both the language and the speaker are analyzed. This is because the former influences the interpretation of the discourse.

Derrida goes beyond this idea and argues that the relationships are not static but dynamic (Wallemacq et Jacques, 2001). Linstead (quoted in (Wallemacq et Jacques, 2001)) writes : *"Central to Derrida's thought is the recognition that language always embodies a relationship of power between terms, one being used rather than another possible term in any text"*. The language according to Derrida is not a set of words that exist peacefully, some of them being in association or opposition with others. Instead, the set of relationships of the language undergoes perpetual tension. In order to integrate this dynamic dimension into the relationships system, Derrida devises the term *"differance"*. It finds its etymology in the French language and more precisely from the French verb *"différer"*, which carries the meaning of both English verbs *"differ"* and *"deffer"* (Wallemacq et Jacques, 2001).

The idea of analyzing relationships from a language rises an essential question : what is exactly the language? The most intuitive answer to this question would be a language such as English, or French. However, Wallemacq has shown that *"organizations secrete their own language"* (quoted in (Wallemacq et Jacques, 2001)). Thus, two people who speak French won't necessarily express themselves within the same language. Furthermore, the language may be, and is most likely, a composition of several languages. For instance, a French speaker may work in an organization and live within a family. The family and the organization secrete their own language, on top of the French language.

The existence of a language above the discourse and its influence on the interpretation by the listeners is well illustrated with a practical example by Critchfield in (Critchfield, 2017). This article deals with the scenario of a behavior analyst who must explain the services he can offer to parents who have a child diagnosed with autism. More precisely, the term of interest is *extinction*.
In the domain of behavior analysis, extinction is defined as *"a procedure used in Applied Behavioral Analysis (ABA) in which reinforcement that is provided for problem behavior (often unintentionally) is discontinued in order to decrease or eliminate occurrences of these types of negative (or problem) behaviors"*[1]. However, this is only valid in the language of behavior analysts. Consequently, the child's parents who do not know this language won't interpret the term *extinction* the same way as the behavior analyst does. This aforementioned phenomenon is known by structuralists as the *sense on the rebound*.
Under the assumption that the child's parents speak English and putting aside the other languages they express themselves in, looking at the WordNet (Miller,

[1] https://www.special-learning.com/article/extinction (accessed May 6, 2017)

1995) relationships involving the term *extinction* would be fairly accurate way to understand how they perceive this term. Visuwords[2] is an access-free online visualization tool that allows visualizing the relationships involving a chosen term in the WordNet database. It is the tool used by Critchfield in (Critchfield, 2017). Figure 4.3 shows the Visuwords diagram for the term *extinction*.

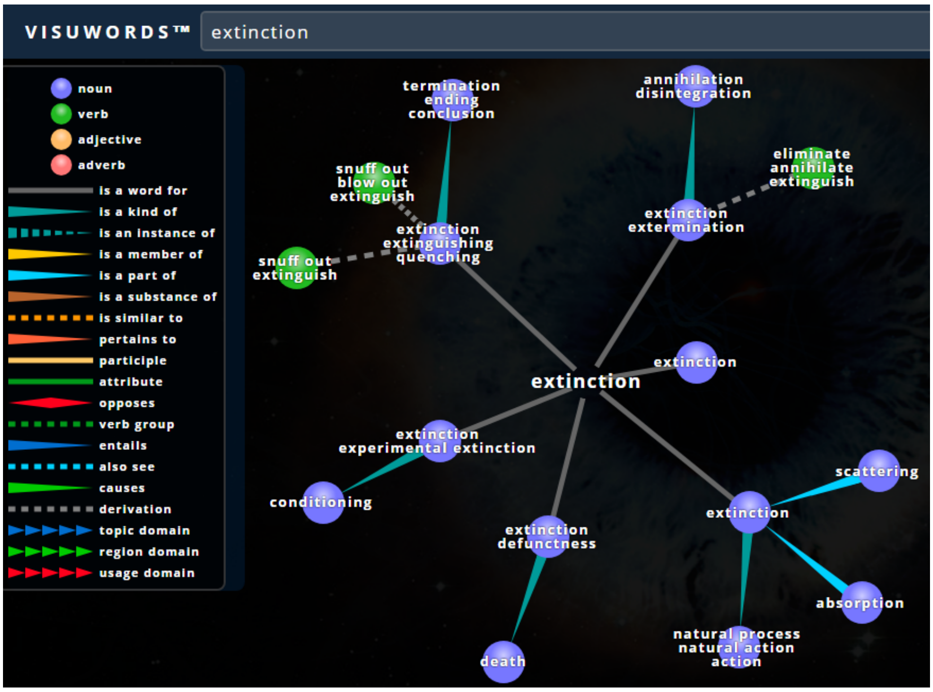


Figure 4.3: Visuwords diagram for the term *extinction*

Figure 4.3 indicates that *extinction* evokes rather negative ideas to the child's parents (annihilation, extermination). Consequently, the behavior analyst should receive quite a horrified look from the child's parents, although he is only offering them help to deal with the child's autism. As Wallemacq and Jacques would say, the behavioral analyst is *"caught in the power of words"* (Wallemacq et Jacques, 2001).

As suggested in (Critchfield, 2017), the solution for behavior analysts to avoid this issue is to use tools like Visuwords to observe how non-experts would interpret their words. This sums up to expressing themselves in the same language as the non-experts, subsequently giving up their expert language.

Overall, the key concepts of the structural approach can be summarized as follows :

- Discourse as a semantic field
  A word is not defined by itself, but by its association and opposition relationships with other words. The meaning of the discourse is relevant,

[2] https://visuwords.com/extinction (accessed May 6, 2017)

however the meaning of words taken individually is of no interest. In order to understand the meaning of a discourse, one must uncover the set of relationships the speaker is situated in.

- The page is never blank
  The discourse lays on top of a language which has its own set of relationships.

- The speaker is caught up in the power of words
  The speaker does not have full control over the meaning of the discourse. The relationships set of the language influences the interpretation of the discourse and threatens the relationships explicitly defined in the discourse.

- Differance
  The words in the set of relationships are in a state of perpetual tension.

In the next section these fundamental concepts are looked at from the angle of visualization in order to see what kind of visualizations are convenient for assisting a user in the structural approach.

4.2 Evaluation Grid

(Wallemacq et Jacques, 2001) provides three high-level guidelines for conceiving adequate representations of semantic fields. They are taken into account together with the concepts discussed in the previous section to build the evaluation grid.

Guideline 1 : *"The properties of the space of representation offered to the user must be homologous to those of structural analysis."*

Four properties that the space of representation should satisfy are given in (Wallemacq et Jacques, 2001). Those are :

- Relational : represents relationships between elements of the text
- Non-additive : when an element changes, it affects all the structure of the representation and not only the changed element and the directly connected ones
- Synchronic : the text order doesn't matter
- Englobing : the text is inside the language

From there, a trivial criterion can be derived. The visualization should represent a set of associations and oppositions relationships. The basic unit of analysis is not an individual word, but a relationship between two words.

Guideline 2 : *"Returning to the analogies used by structuralists and post-structuralists, one appears particularly apt with regard to the structuralist mode of interpretation - the landscape."*

The reasons why the landscape is especially convenient are given in (Wallemacq et Jacques, 2001) :

- The landscape prevents considering words individually. The position of a word will be considered relatively to other words that are further, or at a different altitude in the landscape.
- The landscape has an immersing side, which is suitable to illustrate that the speaker is situated within the semantic field and not outside of it.

Also, the landscape metaphor is widely used and speaks to everyone. Thus, users can easily benefit of and understand this representation because they can transpose it to something they know.

However, Derrida's concept of differance implies that the visualization should reflect the tension perpetually underwent by the words in the set of relationships. In that respect, the landscape metaphor may expose its limitations, since a landscape as commonly conceived is rather static than dynamic. In regards to Derrida's differance, the field of force (as mentioned in (Wallemacq et Jacques, 2001)) may prove to be a more suitable representation.

Guideline 3 : *"Visualization is not the dressing-up of an interpretation, but is an active part of the interpretation itself."*

This guideline indicates that the visualization must help the analyst to gain knowledge over the meaning of the discourse. This implies that the visualization should provide interaction since interactive visualizations are more insightful than static ones. The interaction functionality must obviously be consistent with the principles of structural analysis (see guideline 1).
The kind of knowledge discovering that an interactive representation can provide is the observation of how the semantic field reacts to a change. Examples of relevant changes a user could make to the semantic field are :

- Add a relationship
- Delete a relationship
- Choose another language set

Guideline 3 also illustrates that the representation has assumptions which influence the interpretation.

The idea that the page is never blank implies that a base of knowledge is needed. The speaker's set of relationships can be uncovered by merely analyzing the text but the language's set must be found elsewhere. There are two possibilities to obtain it :

- Use an online synonyms/antonyms dictionary
  <u>Advantage</u> : provides a comprehensive set of relationships
  <u>Drawback</u> : comprehensive data available only for widely used languages (e.g. millions of speaker)

- Merge the speaker's sets of previously analyzed texts
  Advantage : it is possible to build a set for a less widely used language such as an organizational language
  Drawback : need of a great amount of discourses to provide a fairly comprehensive set

The visualization must thus be able to represent the relationships of the speaker and the language, but not only. The aforementioned example from (Wallemacq et Jacques, 2001) then analyzes the relationship between *management* and *economics* (see Figure 4.1 and 4.2) shows that relationships can be uncovered by symmetry. These are referred to as *implicit* relationships. The visualization should therefore be able to uncover and represent the implicit relationships.

In a nutshell, a visualization that is to assist a structural analyst should satisfy the following criteria :

- Criterion 1 : the visualization has an immersing side that indicates that the speaker is situated within the semantic field
- Criterion 2 : the visualization represents a set of associations and oppositions relationships between terms. The basic unit of analysis is not an individual term, but the relationship between two terms
- Criterion 3 : the visualization uncovers and represents the implicit relationships according to the rules of the structural analysis
- Criterion 4 : the visualization reflects the fact that the words of the set are in tension
- Criterion 5 : the visualization provides interaction possibilities that allows observing the impact on the semantic field of a change such as the adding or the removal of relationships
- Criterion 6 : the visualization uses an external relevant base of relationships to represent the semantic field of the language

There is one limit to take into account for the first criterion. The visualizations are displayed on a 2D screen and could thus be deemed not truly immersing. However, for the needs of the evaluation, more flexibility on the immersing criterion is permitted.

4.3 Evaluation of Existing Techniques and Tools

The goal of this section is to conduct an evaluation of the visualization techniques mentioned in this thesis. More precisely, these are TileBars (Hearst, 1995), TextArc (Paley, 2002), Arc Diagram (Wattenberg, 2002), DocuBurst (Collins *et al.*, 2009), Phrase Net (Van Ham *et al.*, 2009), Word Tree (Wattenberg et Viégas, 2008), Semantic Graph (Rusu *et al.*, 2009), VarifocalReader (Koch *et al.*, 2014), and the four visualizations available in STAVIZ.

The second and the third criteria are specific to structural analysis. Since the visualization techniques presented in the state of the art were not designed specifically for structural analysis, those criteria were not taken into account. The evaluation was thus conducted considering criteria 1, 4, 5, and 6.

The basic unit of analysis of the TileBars visualization is the term. TileBars represents the frequency of a term throughout documents and allows co-occurrence comparisons between terms. The representation it uses achieves low immersiveness and is static. As a result, no tension between terms is reflected and none of the sought after interaction possibilities are offered. Moreover, TileBars does not use any external base of relationships. With regard to the evaluation grid, TileBars is not suited to assist an analyst in the structural approach. Nonetheless, it offers a way to visualize the frequency of terms text chunk by text chunk. It could thus be convenient to give an overview of the important terms in a text, but not to perform structural analysis strictly speaking.

In the view of the evaluation grid, TextArc is similar to TileBars. It is a static visualization and offers none of the desired interactions, nor does it uses an external relationships base. However, it shows three noteworthy differences. Firstly, the terms displayed in a TextArc are not chosen by the user. Secondly, TextArc does not consider a numerical frequency by big chunk but rather a boolean occurrence by sentence. Thirdly, TextArc achieves better immersiveness than TileBars.

Again, the Arc Diagram shows compliance to the evaluation grid similar to TileBars and TextArc. It is also a static visualization that shows the same absence of tension between terms, of interaction, and of external relationships base. The unit of analysis is the relationship, but it is not the relevant type of relationship. It is a co-occurrence relationship between a term and the same term further in the text. However, even if it is not suited to help with structural analysis strictly speaking, it can prove to be useful to get a quick overview of a text in the same way as TileBars and TextArc.

The evaluation shown that TileBars, TextArc, and the Arc Diagram are not suitable candidates for structural analysis. However, they illustrate how convenient they can prove to give a quick overview of the terms distribution in a text. They provide more knowledge than a mere count of the occurrences in a text. It may be helpful to some analysts to have such information about the most frequent terms before proceeding to structural analysis, or event during the analysis to know the distribution of a chosen term.

DocuBurst displays hyponymy relationships as a fairly immersing sunburst diagram. It shows a tiles visualization of the text to indicate the distribution of the terms in the sunburst. It does not provide any relevant interaction on the relationships and does not reflect any tension between terms. In this regard, it is similar to the three previous techniques. However, it uses an external relationships, namely WordNet, from which it retrieves the hyponymy relationships. Ultimately, its static nature, the lack of interaction, and the fact that hyponymy relationships are not the ones of interest make DocuBurst a non-suitable technique for structural analysis. However, DocuBurst supports the interest of the tiles visualization to show the frequency distribution and illustrates well how an external relationships base can be used to add information over a text.

Phrase Net represents relationships between two terms as a directed graph, which achieves great immersiveness considering the 2D screen limitation. The relationships it represents are of type A *word* B, with A and B two terms of the text and *word* a word of the user's choosing. The relationship exists if the string "A *word* B" exists in the text. In the context of structural analysis, this way to do has limitations. Since there is no formal way to describe a relationship (Wallemacq *et al.*, 2004), defining a relationship type by choosing a "link word" will unlikely achieve good precision in the relationships discovery. Again, Phrase Net does not offer the desired interactions and tension reflection, and it does not rely on any external relationships base. However, since Phrase Net shows the terms frequency, it can be used with pre-encoded relationship types such as *is* or *is not* to provide a rich overview of the text.

The Word Tree represents consecutiveness relationships between terms, which would likely show low precision in the relationships discovery for structural analysis. The reason is the informal nature of the relationships (Wallemacq *et al.*, 2004). As for the evaluation grid, the compliance scores are strongly similar to those of the Phrase Net, except that the Word Tree is less immersive. However, the Word Tree manages to keep all the sentences of a text in a more concise way. Thus, even it is not a suitable visualization to represent relationships, it could replace the text as analysis material.

The Semantic Graph represents a text as a set of relationships between a subject, a verb, and an object forming one sentence in the text. As discussed before, such a way of discovering relationships is inaccurate in the context of structural analysis. The results of the application of the evaluation grid are the same as those of the Phrase Net and the Word Tree. However, the Semantic Graph is as immersing as the Phrase Net and it uses WordNet synsets to condense the representation.

VarifocalReader is not evaluated here because it was not made to display relationships in the first place. However, it is noteworthy because it reminds the importance of keeping access to the text. Indeed, visualizations are often displayed without the text and the analyst has to go to another page to see the text. Also, in most cases, brushing and linking between the text and the visualization is not implemented. Maintaining access to the original text and linking it to visualization is particularly relevant in the context of structural analysis. Ultimately, even though VarifocalReader is not a suitable candidate for structural analysis, it is important to mention it because it reminds an essential consideration that few visualization techniques actually take into account.

STAVIZ's word cloud represents the terms frequency in a text. It offers interactions such as moving and hiding a term, but these are not the sought over interaction possibilities. The word cloud displays no relationship, and consequently reflects no tension between terms. Furthermore, it uses no external relationships dictionary. However, it achieves reasonable immersiveness considering the aforementioned 2D screen limitation. As previously said, an overview of the most frequent terms could be useful prior to structural analysis strictly speaking, but Tilebars, TextArc, and the Arc Diagram do it in a more informative way.

STAVIZ's matrix represents the relevant relationships for structural analysis, as well as the implicit ones. However, it achieves low immersiveness, is static, and lacks of interaction possibilities. The matrix can use Wikipedia to infer potentially interesting relationships. However, the type (opposition or association) of the inferred relationships has to be determined by the user and the Wikipedia approach shows penalizing limitations as discussed in Subsection 3.6.2. Thus, criterion 6 is not satisfied since Wikipedia is not a relevant external relationships base. Despite this, the matrix visualization remains useful for structural analysis. Its insensitivity to the relationships set size (Ghoniem *et al.*, 2005) makes it more convenient than its alternatives to show a lot of relationships at once. Moreover, as previously discussed, the matrix speeds up the encoding of relationships.

STAVIZ's chord diagram represents the relevant relationships for structural analysis, as well as the implicit ones, in a more immersing way than the matrix does. As for the other criteria of the grid it has the same compliance as the matrix. It is worth mentioning that the chord diagram does not use any external relationships base. However, the chord diagram shows relationships with less clutter than the node-link diagram and offers a focus interaction that improves its legibility. In this respect, the chord diagram can prove convenient when there is a need for a trade-off between immersiveness and readability. Its usefulness in the present context can nonetheless legitimately be questioned by the presence of the focus interaction in the node-link diagram.

STAVIZ's node-link diagram represents the same relationships as the chord diagram does. It achieves great immersiveness considering the 2D-screen limitation discussed earlier. It provides interactions to remove and add relationships at will. However, it is incapable of reflecting tension between terms despite its dynamic nature and its drag-and-drop interaction. Moreover, it uses no external relationships base. Even though it has weaknesses, the node-link diagram was deemed the most useful technique in STAVIZ.

4.4 Improving STAVIZ

The goal of this section is to propose improvements of STAVIZ based on the lessons learned from the evaluation. In doing so, it provides lines of thought on how to make STAVIZ more suitable for structural analysis. These reflections are then used to draw a paper prototype of what an improved STAVIZ could look like. It is hoped that a refactoring of STAVIZ following the lines of the subsequent subsections will produce a tool that is genuinely helpful for structural analysts.

The improvements proposed for STAVIZ are the following. They are detailed in subsequent dedicated subsections.

- Removing the chord diagram
- Removing the Wikipedia-based relationships suggestion

- Adding a visualization combining a word cloud and tiles to give an informative overview about the frequency of terms
- Improving the node-link in order to satisfy the six criteria of the evaluation grid
- Keeping the text close to the visualization
- Making the whole tool fit into one page

4.4.1 Usefulness of the Chord Diagram

As shown in the evaluation, the usefulness of the chord diagram resides in its well-balanced trade-off between immersiveness and readability. However, it can be discussed whether it is worth proposing a visualization for this purpose only given that the node-link diagram offers a focus interaction.

For this reason, the chord diagram should not be integrated in an improved STAVIZ.

4.4.2 Removing the Wikipedia-Based Relationships Suggestion

The precision of the results observed during the tests as well as the discussed limitations of this approach indicate that the Wikipedia-based relationships suggestion should not be kept.
However, the text-based relationships suggestion is more consistent in the context of structural analysis because it allows uncovering relationships from the text only. It should thus be maintained in an improved version of STAVIZ. It can be combined with an external base such as WordNet whose relevance is shown by DocuBurst, the Semantic Graph, and (Critchfield, 2017). This combination will hopefully allow a comprehensive discovery of the relationships.

4.4.3 Keeping the Text Close to the Visualization

The visualizations in STAVIZ are currently opened in other pages. As a result, the user has to return to the main page to get to the original text. This makes it inconvenient to see the text and a visualization simultaneously. However, VarifocalReader illustrates how convenient it can be to always have the original text on hand. This idea should be reproduced in an improved STAVIZ for this reason. Moreover, it is especially relevant in the context of structural analysis since analysts often need to get back to the text throughout the process.

4.4.4 Making Everything Fit Into One Page

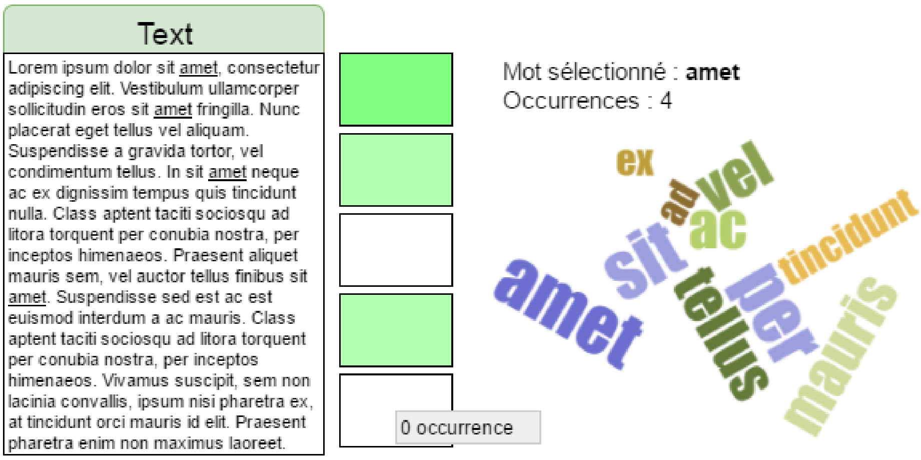
The removal of less useful features and the idea of keeping the text close to the visualization lay the groundwork for a one-page tool. The reasons why an improved STAVIZ should integrate this notion of one-page tool are the following:

- Since STAVIZ is web-based, having the visualizations and the text on several separated tabs could prove confusing if the user already has tabs opened for other uses

- If STAVIZ cannot fit into one page, then it may offer too many features
- Linking and brushing is more informative when the views are displayed simultaneously

4.4.5 Adding a Visualization Combining a Word Cloud and Tiles

The evaluation of TileBars, TextArc, the Arc Diagram, and DocuBurst shows that a terms frequency overview can be more informative than a mere count of the occurrences. The word cloud is useful to give the most frequent terms, but it should be combined with a tiles visualization to show how a given term is distributed in the text.
Figure 4.4 shows a quick sketching made with `https://www.draw.io` of such a visualization. It represents a text, tiles, and a word cloud of the ten most frequent terms in the text. The user can select a term in the cloud and see its number of occurrences in the text. The tiles to the right of the text indicate how a term is distributed. The word cloud was generated with Jason Davie's algorithm[3].



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ullamcorper sollicitudin eros sit amet fringilla. Nunc placerat eget tellus vel aliquam. Suspendisse a gravida tortor, vel condimentum tellus. In sit amet neque ac ex dignissim tempus quis tincidunt nulla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Praesent aliquet mauris sem, vel auctor tellus finibus sit amet. Suspendisse sed est ac est euismod interdum a ac mauris. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus suscipit, sem non lacinia convallis, ipsum nisi pharetra ex, at tincidunt orci mauris id elit. Praesent pharetra enim non maximus laoreet.

Mot sélectionné : **amet**
Occurrences : 4

Figure 4.4: Word cloud combined with a tiles visualization. The term *amet* is selected.

4.4.6 Improving the Compliance to the Grid of the Node-link Diagram

One visualization technique performs better than the others with respect to the proposed evaluation grid : the node-link diagram from STAVIZ. It satisfies four criteria out of six, namely criteria 1, 2, 4, and 5. This subsection will thus focus on this visualization and provide lines of thought on how the two remaining criteria could be satisfied. When relevant, solutions for improving the compliance to satisfied criteria are provided.

[3] https://www.jasondavies.com/wordcloud/ (accessed May 14, 2017)

Improving Compliance to Criterion 1

Although the node-link diagram has a fairly immersing side, the recent technological advances offer an immense range of possibilities for visualizing semantic fields. In particular, virtual reality could be exploited to create a truly immersing representation which would potentially allow more powerful reasoning over the semantic field. The idea of the speaker expressing himself within the field would be genuinely materialized. This yields promising possibilities of improvements for STAVIZ.

Improving Compliance to Criterion 2

STAVIZ used only one rule to uncover implicit relationships. With the help of structural analysts, other relevant rules could be formalized and subsequently implemented in STAVIZ.

Satisfying Criterion 3

One of the main remarks given by the members of the EFFaTA-MeM project on STAVIZ at the end of the internship regards the perception of the tension in the visualization. The node-link visualization attempts to reflect the tension between terms, but it shows a gap for the opposition relationships.

The tug of wars metaphor helps grasping this notion of tension. The tug of wars is a strength competition between two people, or two groups of people. They pull on opposite ends of a rope and the goal of the game is pull the rope stronger than the opponent to force them to reach a line drawn on the floor. When transposing a tug of wars game to a relationship on the node-link diagram, the competitors become the nodes and the rope becomes the link.
The reason why this metaphor is insightful resides in the fact that the terms on the node-link diagram should behave like tug of wars competitors. When a team pulls the rope in a tug of wars match, the other team pulls in the opposite direction, subsequently subjecting the rope to perpetual tension. Applying this metaphor to the node-link diagram involves adapting the effect of the drag-and-drop interaction. Figure 4.5 explains the adaptation needed. It shows an initial layout, a drag-and-drop interaction being triggered, how the diagram currently reacts, and how the diagram should react to the event.

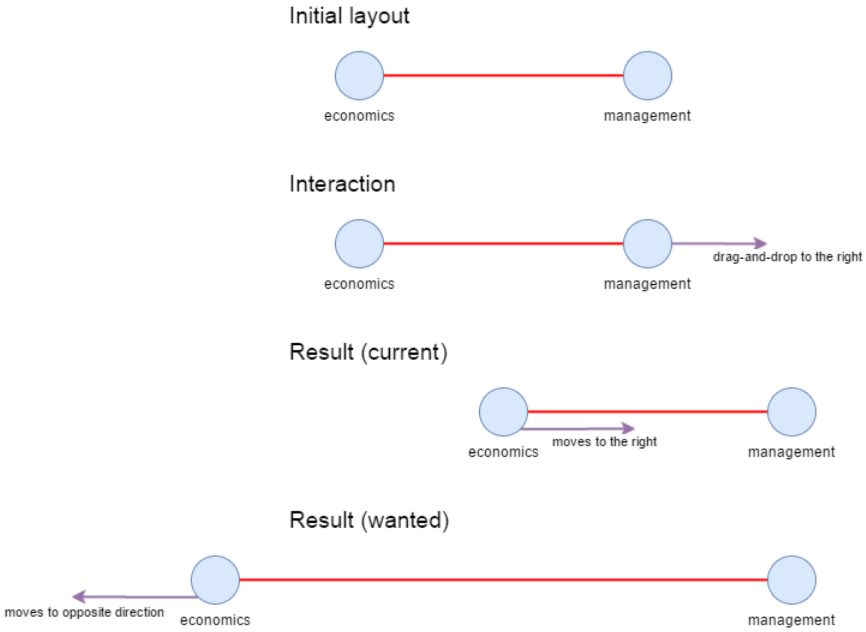


Figure 4.5: Adaptation needed for the node-link diagram drag-and-drop interaction. The blue disks represent terms, the red line represents an opposition relationship, and the purple arrows represent the interaction/reaction described by its label)

As far as implementation is concerned, it is equivalent to modify the forces set on the d3 visualization initialization. Three forces are involved when a force layout is generated with D3.js, namely :

- Gravity : defines whether the nodes will tend to attract (gravity greater than 0) or repel (gravity lesser than 0) each other. All the nodes undergo the same gravity force.
- Link distance : determines the initial distance between two given linked nodes, that is, the link length. After a change on the layout, the link tends to recover this length. The link distance can be set for each link separately.
- Link strength : defines how a node of a given link will react subsequently to a disturbance on the other. The link strength can be set for each link separately.

The problem with the previous implementation is that the link strength force was not defined. Thus, the default value (force = 1) was set. This explains why a node followed its opposite when the latter was dragged. The improved Javascript code responsible for the force settings is given below. It works with the version 4 of D3.js.

```
var force = d3.layout.force()
  .size([width, height]) // Layout size
```

```
    .charge(-1000) // Gravity
    .linkDistance(function(link){
      return link.value*40 - 10; // Link distance
    })
    .on("tick", tick);

  force.linkStrength(function(link){ // Link strength
    if(link.value < 5){
      return 1; // Default value (dragging a node will make the other
          move in the same direction) for association relationships
    }
    return -0.01;  // Repelling force for opposition relationships
  });
```

When the settings of one force are modified it may have an unexpected effect on the other forces. For example, modifying the link strength can have an impact on the link distance. Since this force layout was a first experience with D3.js forces, trial an error was necessary to find a suitable equilibrium of the forces. The obtained results are a substantial improvement of the previous implementation, yet more experience with D3.js could potentially give better results.

Finally, it is worth noting that bettering the drag-and-drop interaction would improve the relevance of the interaction functionality, consequently strengthening the compliance to the fifth criteria.

Satisfying Criterion 6

The idea that speaker expresses a discourse within a preexisting set of relationships between terms is missing in STAVIZ. It refers to the aforementioned *englobing* property of the space of representation. Conceptually, solving this issue is simple : it is equivalent to add this preexisting set, namely the relationships set of the language. However, as briefly discussed in Section 4.2, the issue of finding this set is not a simple one for the following reasons :

- Language multiplicity : the language is not unique, for example, an American man working as a llama raiser expresses himself withing the English language and the language of his organization, and potentially others as well

- Language variability : the language is not necessarily fixed, this American man does not necessarily expresses himself withing its organization's language when he speaks to his family

- Language source : dictionaries containing a fairly comprehensive set of relationships exists only for widely used languages such as English or French, but not for organizational languages

These three issues need to be addressed so that STAVIZ satisfies the sixth criterion.

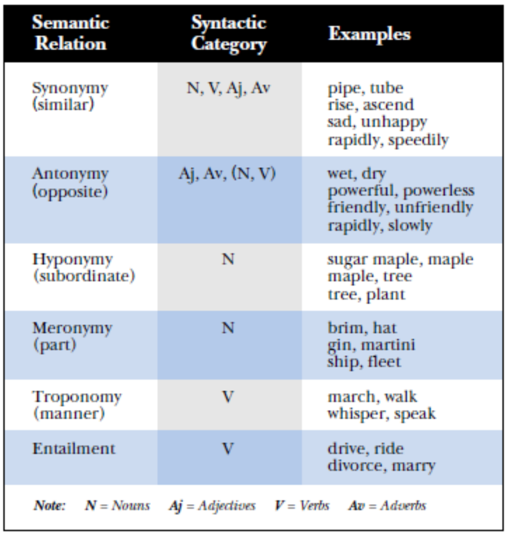
Language Multiplicity Expressing oneself within several language is equivalent to expressing oneself within one language that is the union of all the languages. Given that languages here are thought of as relationships sets, the language multiplicity issue is solved by considering one relationships set that is the union of the respective sets of all the languages. STAVIZ should provide a user-friendly interface to create such union sets.

Language Variability The language variability issue can be solved by adding a new interaction feature to STAVIZ. It would allow the user to modify the composition of the language relationships union set and see the effect on the visualization.

Language Source This issue leads to consider several types of languages. For the sake of simplicity, only two types are considered here :

- General use language : widely used language, everyday language
  Examples : English, French, Finnish
- Organizational language : language used by a restrained number of people
  Examples : behavioral analysts language, language spoken by workers of a llama raising industry

Online dictionaries provide fairly comprehensive relationships sets for many general use languages. A well-known and widely used example for the English language is WordNet (Miller, 1995). WordNet is a database containing relationships between synsets (sets of cognitive synonyms). There are more than 200,000 relationships in the WordNet database[4]. There are different types of relationships in WordNet. They are listed in Figure 4.6.



Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	N, V, Aj, Av	pipe, tube rise, ascend sad, unhappy rapidly, speedily
Antonymy (opposite)	Aj, Av, (N, V)	wet, dry powerful, powerless friendly, unfriendly rapidly, slowly
Hyponymy (subordinate)	N	sugar maple, maple maple, tree tree, plant
Meronymy (part)	N	brim, hat gin, martini ship, fleet
Troponomy (manner)	V	march, walk whisper, speak
Entailment	V	drive, ride divorce, marry

Note: *N = Nouns* *Aj = Adjectives* *V = Verbs* *Av = Adverbs*

Figure 4.6: Relationship types in WordNet (taken from (Miller, 1995))
Note : the synonymy relationships is for words, a synset is a set of words linked by a synonymy relationship

[4] http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html (accessed May 7, 2017)

In the case of STAVIZ, the relationships of most interest are the synonymy and the antonymy. They respectively correspond to the association and opposition relationships. Thus, the relationships set for the English language is composed of all the synonymy and antonymy relationships in WordNet.

As mentioned in (Critchfield, 2017), equivalents of WordNet exist for other general use languages as well.

Concerning the French language, CRISCO has published a dictionary comprising 200,000 relationships (Manguin, 2005). It provides synonymy as well as antonymy relationships. CRISCO's dictionary is available online[5]. Moreover, Manguin noted a strong progression of the numbers of queries on the synonyms dictionary, showing a growing interest in this kind of service (Manguin, 2005). If this trends still carries on today, this gives great hope for always more comprehensive dictionaries.

The language source issue can be solved by relying on online dictionaries for general use languages, but no such dictionary exist for organizational languages. These language are much less formalized and have few users.

This issue can be solved to a certain extent by developing an analysis sharing platform on top of STAVIZ. In turn, the structural analysts that work on discourses from the same organization could combine the relationships dictionary resulting from their respective analysis. The resulting dictionary would represent the set of relationships of the organizational language. However, this approach has drawbacks :

- There are also relationships from other organizational languages in the relationships dictionary. Filtering the relevant ones would be tedious if not impossible, and not filtering would result in many noise
- A lot of text analysis would be necessary in order to build a duly comprehensive set of relationships

On account of these drawbacks, building a comprehensive relationships set for an organization remains an open issue in the context of STAVIZ. Yet, the sharing platform would yield added value. Organizational language relationships put aside, such a platform would certainly help to popularize the structural approach.

4.5 Prototyping an Improved Version of STAVIZ

This section proposes a paper prototype of a an improved version of STAVIZ. It is based of the improvements lines of thought provided in the previous section. The objective here is not to define how the improved STAVIZ should look like and what functionality it should offer. On the contrary, the goal is to give an example of an improved version of the software for further reflection and to give inspiration for prospective future work.

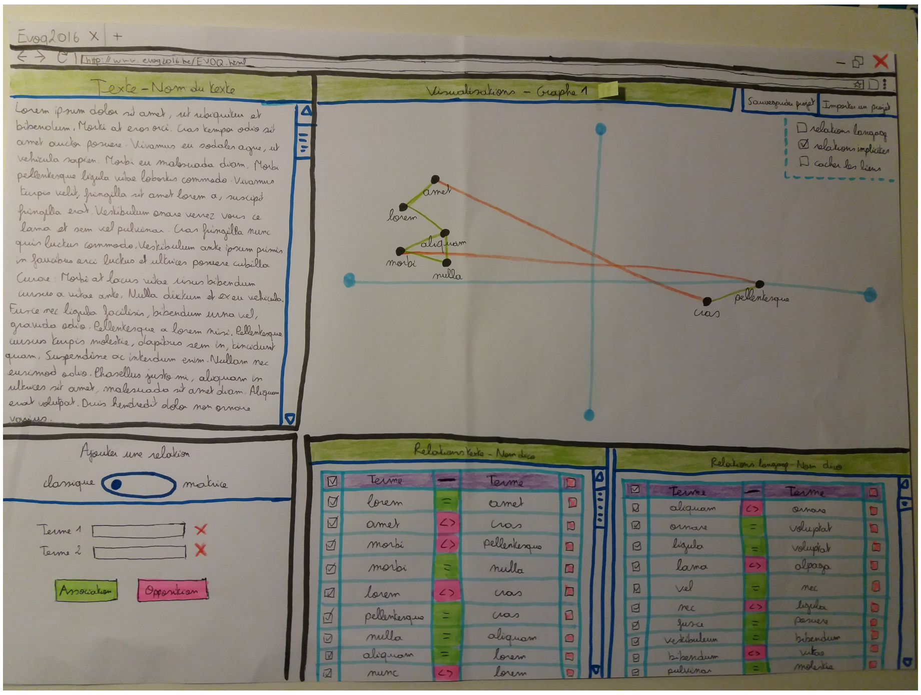
[5] http://www.crisco.unicaen.fr/des/synonymes/ (accessed May 7, 2017)

Furthermore, it is noteworthy to say that the proposed prototype was not evaluated by any prospective end users. Conducting such an evaluation before considering any implementation is strongly recommended.

The technique used was rapid paper prototyping. The choice of this technique is explained by its low time consumption nature and by its ability to communicate a fairly rich design vision. The sketching is not extensively detailed since the objective is to provide inspiration rather than a finished prototype. Pictures of the sketching are provided in the following subsections.

4.5.1 Main Page and Encoding

Figure 4.7 shows the main page of the improved STAVIZ.



Lorem ipsum dolor sit amet, ut ubiquiten et bibenolum. Morbi at eros orci. Cras tempor odio sit amet auctor posuere. Vivamus eu sodales augue, ut vehicula sapien. Morbi eu malesuada diam. Morbi pellentesque ligula vitae lobortis commodo. Vivamus turpis velit, fringilla sit amet lorem a, suscipit fringilla erat. Vestibulum ornare verez vous ce lama et sem vel pulvinar. Cras fringilla nunc quis luctus commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae. Morbi at lacus vitae risus bibendum cursus a vitae ante. Nulla dictum et ex eu vehicula. Fusce nec ligula facilisis, bibendum urna vel, gravida odio. Pellentesque a lorem nisi. Pellentesque cursus turpis molestie, dapibus sem in, tincidunt quam. Suspendisse ac interdum enim. Nullam nec euismod odio. Phasellus justo mi, aliquam in ultrices sit amet, malesuada sit amet diam. Aliquam erat volutpat. Duis hendrerit dolor non ornare varius.

Relations texte - Nom dico

☑	Terme	—	Terme	🟥
☑	lorem	=	amet	🟥
☑	amet	<>	cras	🟥
☑	morbi	<>	pellentesque	🟥
☑	morbi	=	nulla	🟥
☑	lorem	<>	cras	🟥
☑	pellentesque	=	cras	🟥
☑	nulla	=	aliquam	🟥
☑	aliquam	=	lorem	🟥
☑	nunc	<>	lorem	🟥

Relations langage - Nom dico

☑	Terme	—	Terme	🟥
☑	aliquam	<>	ornare	🟥
☑	ornare	=	volutpat	🟥
☑	ligula	=	volutpat	🟥
☑	lama	<>	alpaga	🟥
☑	vel	=	nec	🟥
☑	nec	<>	ligula	🟥
☑	fusce	=	posuere	🟥
☑	vestibulum	=	bibendum	🟥
☑	bibendum	<>	vitae	🟥
☑	pulvinar	=	molestie	🟥

Figure 4.7: Prototyping a proposition of improved STAVIZ : main page

Significant differences with the current version of STAVIZ are noticeable at first sight :

- The terms dictionary was removed. The reason for this choice is that the structural analysis uses the relationship as unit of analysis. Hence showing only the terms in a table is irrelevant.
- The relationships dictionary takes less place, which allows placing the relationships encoding module next to it.
- There are two relationships dictionaries : one for the text and one for the language.

- The node-link diagram is presented directly on the main page. The goal is to keep the relations encoding and the text next to it. It is more convenient than working on separate pages and allows a more efficient linking and brushing.

As shown in Figure 4.8, the matrix relationships encoding was preserved. However, the matrix interface may appear too small if the screen resolution is low. Thus, a resize feature should be implemented to reduce the size of the text, the visualization, and the relationships dictionary area in order to allocate more space to the matrix encoding interface.

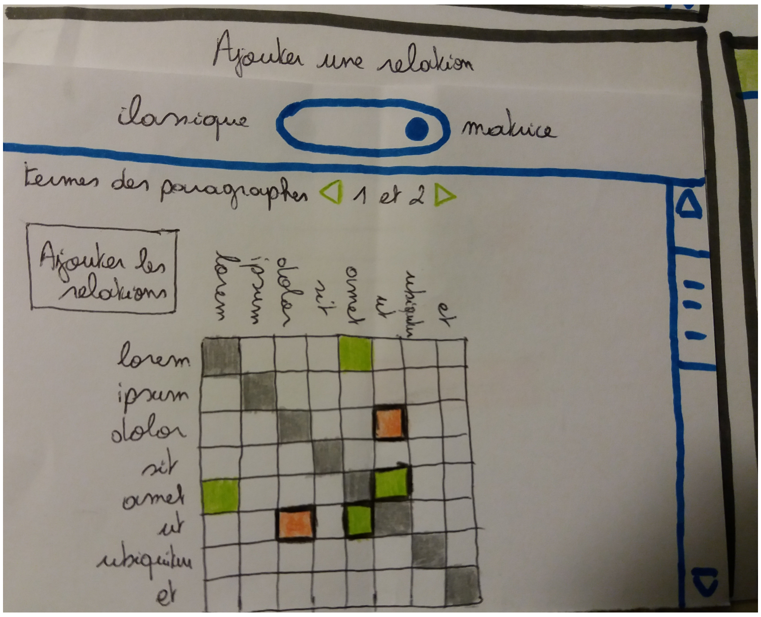
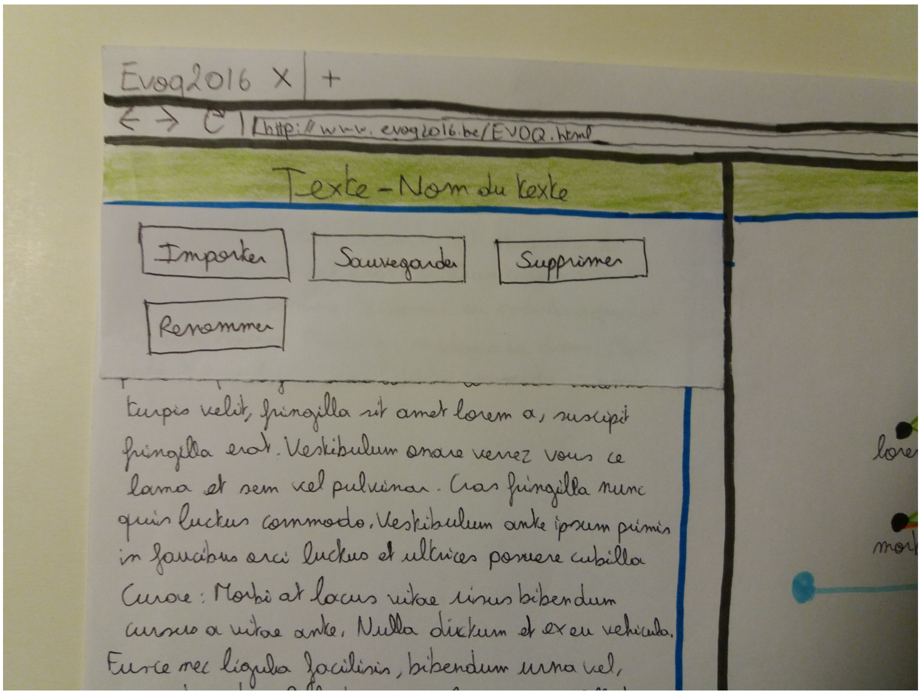


Figure 4.8: Prototyping a proposition of improved STAVIZ : matrix relationships encoding

Moreover, this main page is the only page in the proposed solution.

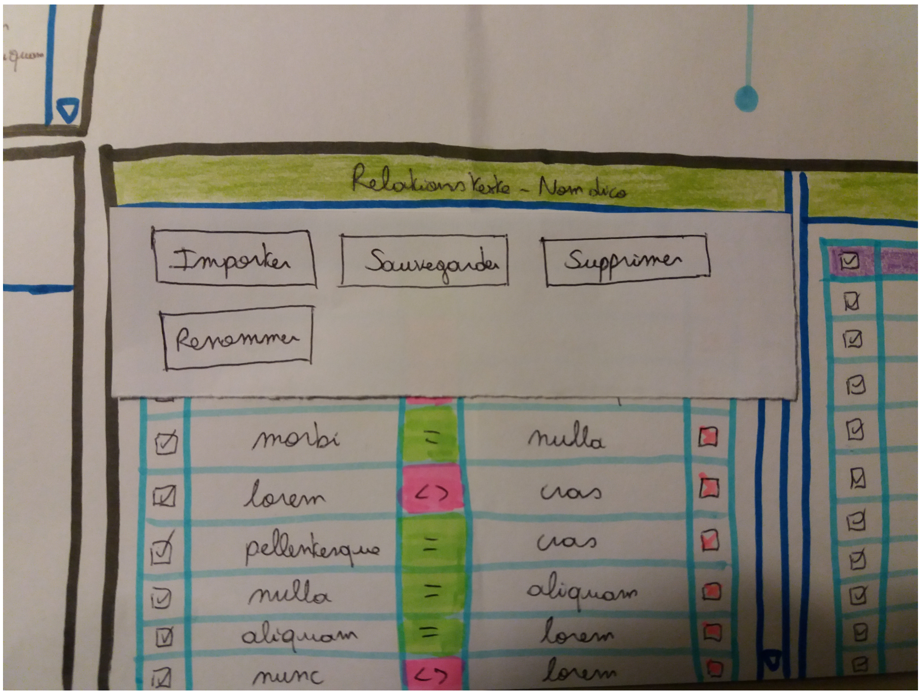
4.5.2 Menus

Since there is a dedicated place for the visualizations in the main page, the menus of the text and of the relationships dictionary have been simplified. The access buttons to the visualizations were moved to a menu dedicated to visualizations. As a result, the menus are more logically organized and are substantially simplified. Figure 4.9 (resp. 4.10, 4.11) shows the menu for the text (resp. the relationships, the visualizations).



turpis velit, fringilla sit amet lorem a, suscipit fringilla erat. Vestibulum ornare venez vous ce lama et sem vel pulvinar. Cras fringilla nunc quis luctus commodo. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae: Morbi at lacus vitae risus bibendum cursus a vitae ante. Nulla dictum et ex eu vehicula. Fusce nec ligula facilisis, bibendum urna vel,

Figure 4.9: Prototyping a proposition of improved STAVIZ : text menu



☑	morbi	=	nulla	🟥
☑	lorem	<>	cras	🟥
☑	pellentesque	=	cras	🟥
☑	nulla	=	aliquam	🟥
☑	aliquam	=	lorem	🟥
☑	nunc	<>	lorem	🟥

Figure 4.10: Prototyping a proposition of improved STAVIZ : relationships menu

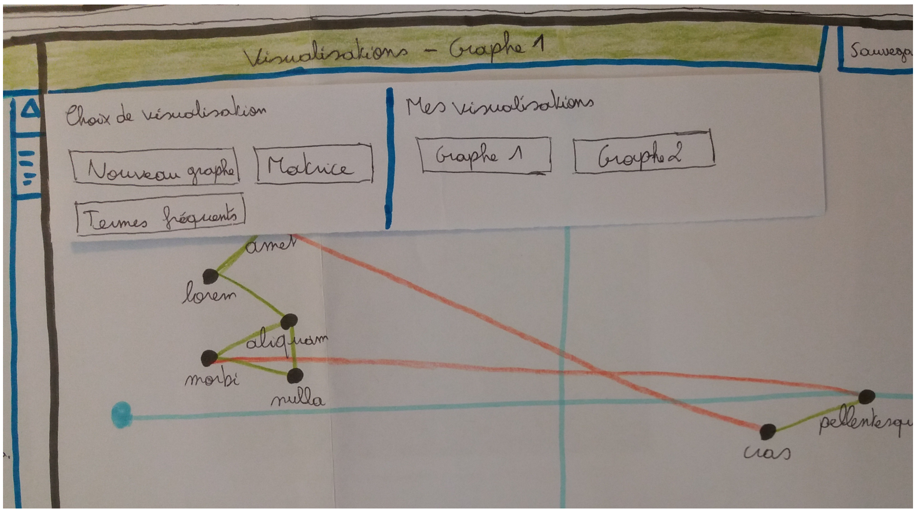
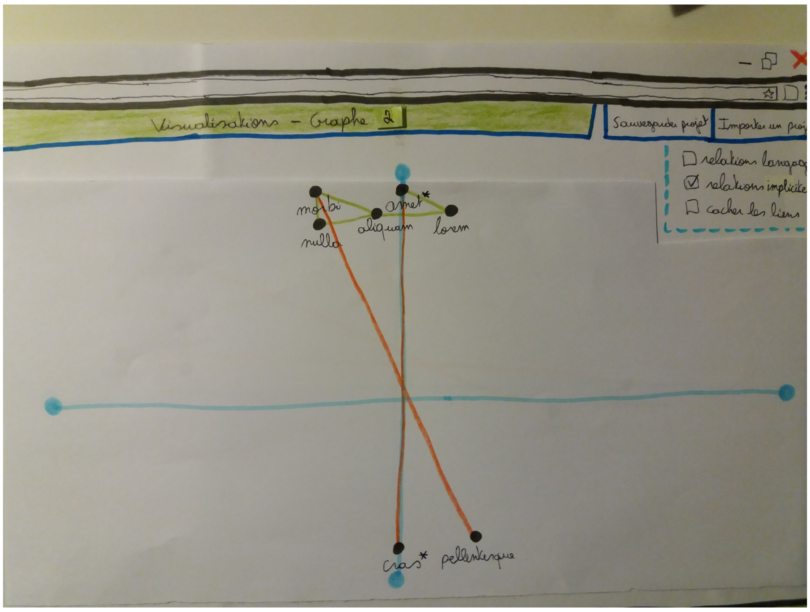


Figure 4.11: Prototyping a proposition of improved STAVIZ : visualizations menu

4.5.3 Visualizations

The visualizations menu on Figure 4.11 highlights several differences with the current version of STAVIZ :

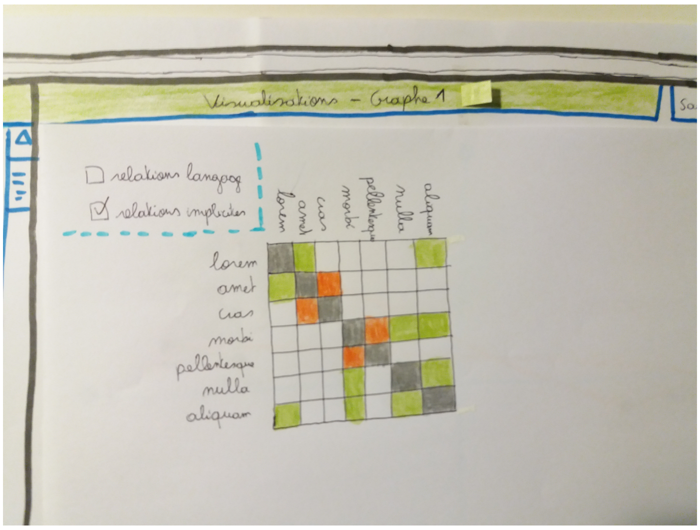
- The chord diagram was removed for the reason highlighted by its evaluation.
- There is no *"word cloud"* button anymore. The word cloud was combined with a tiles visualizations as in Figure 4.4. The resulting visualization was named *frequent terms*.
- The user can create visualizations and manage them. In Figure 4.11 the user has created two node-link diagrams and can switch from one to the other. Figure 4.12 show the second node-link diagram, which is displayed instead of the first one when the user selects it.



□ relations langage
☑ relations implicites
□ cacher les liens

Figure 4.12: Prototyping a proposition of improved STAVIZ : second node-link diagram

Matrix Visualization The matrix visualization is the same as in the current STAVIZ and it serves the same purpose. Since the proposed solution fits in one page, the matrix is displayed over the node-link diagram and the relationships dictionaries as shown in Figure 4.13. Finally, a filtering interaction was added. It allows choosing whether to show the implicit relationships and the relationships of the language dictionary.



□ relations langage
☑ relations implicites

	lorem	amet	cras	morbi	pellentesque	nulla	aliquam
lorem	▨	🟩					🟩
amet	🟩	▨	🟥				
cras		🟥	▨				
morbi				▨	🟥	🟩	🟩
pellentesque				🟥	▨		
nulla				🟩		▨	🟩
aliquam	🟩			🟩		🟩	▨

Figure 4.13: Prototyping a proposition of improved STAVIZ : matrix visualization

Node-link Visualization The interaction functionality of the node-link diagram has been simplified.

Firstly, the relationships adding and removing feature was suppressed because it became redundant with the delete feature in the relationships table and by the relationships encoding module. The reason for this redundancy lies in the single-page logic of the proposed solution. Since the node-link diagram is on the same page as the other modules, the aforementioned features become redundant.

Secondly, a filtering feature was added to the node-link diagram, allowing the user to choose the relationships to display on the diagram.

Figure 4.14 shows the interaction features for the nodes, still available on a context menu displayed after a right click on a chosen node.

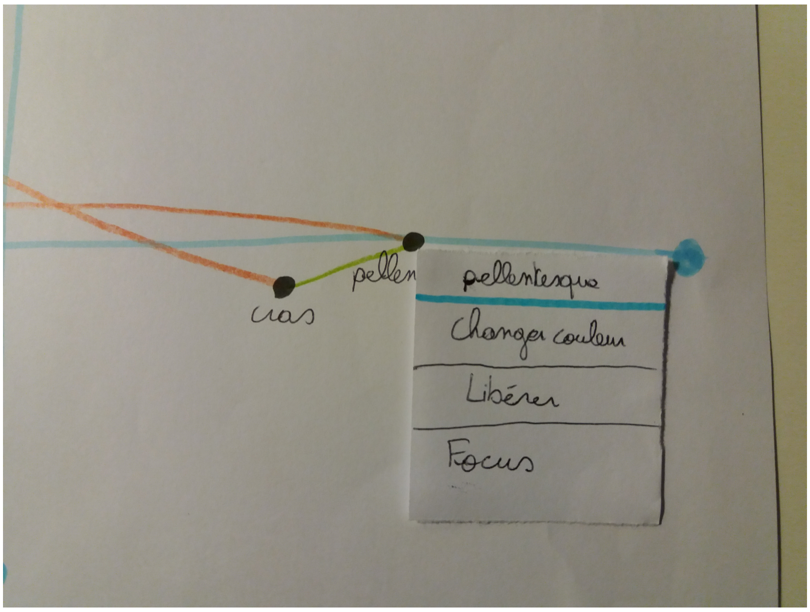


Figure 4.14: Prototyping a proposition of improved STAVIZ : context menu of the node *pellentesque*

Frequent Terms Visualization The frequent terms visualization allows visualizing a chosen number of frequent terms on a word cloud. The user can then select terms in the cloud and observe how they are distributed in the text through the tiles visualization.

Figure 4.15 shows the frequent terms visualization with three selected terms.

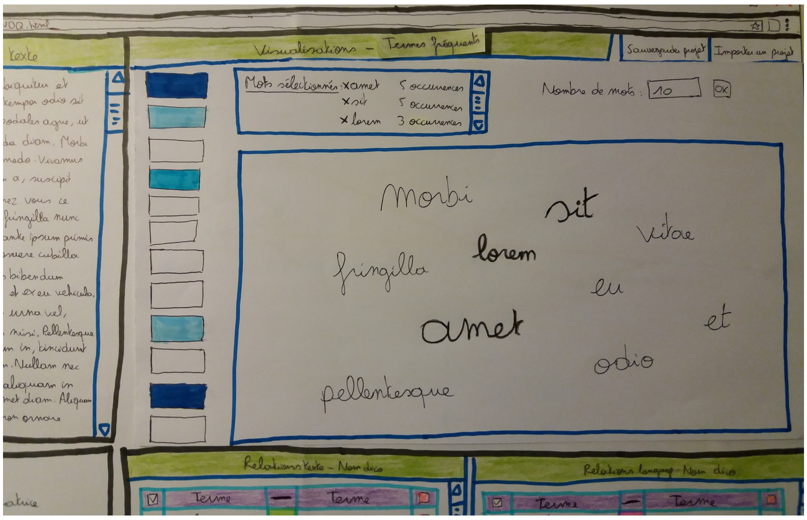


Figure 4.15: Prototyping a proposition of improved STAVIZ : frequent terms visualization

A hover menu shows the number of occurrences for each select term for a given tile (Figure 4.16).

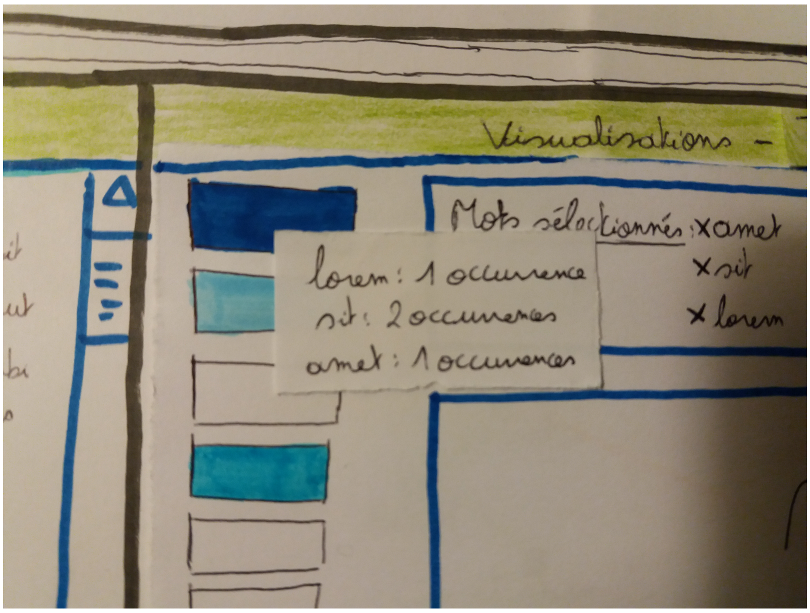


Figure 4.16: Prototyping a proposition of improved STAVIZ : hover menu of the first tile. There are one occurrence of the term *lorem*, two occurrences of the term *sit*, and one occurrence of the term *amet* in the text chunk represented by the first tile.

Chapter 5

Conclusion

The interest of this thesis lies in the sometimes unsuspected crossing of two disciplines : sociology (structural discourse analysis) and computer science (text visualization). More precisely, its objective is to understand how text visualization techniques can help sociologists in the structural analysis process.

This problematic was broken down into two research questions addressed in this thesis :

- Which existing visualization techniques and tools could be relevant to structural analysis?
- What makes a visualization tool suitable to help structural analysts?

The second research question was explored during an internship at the University of Namur. The result of this work is STAVIZ, a software designed to help structural analysts in their work.
The first research question was answered by conducting a state of the art of the visualization techniques and tools that represent relationships between text elements.
Based on this work, on the insights and the feedback gathered during the development of STAVIZ, and on a further review of the key concepts of structural analysis, this thesis answered its second research question by elaborating a six-criteria evaluation grid. It can either be used as a list of requirements for the development of a visualization tool for structural analysts or as a methodology to assess the usefulness of an existing tool for those analysts.

The proposed evaluation grid was then applied to the reviewed text visualization tools and to STAVIZ. The result of this process was a detailed set of lines of thought on how to make STAVIZ more relevant to structural analysis. The objective was to prepare the ground for the numerous possible future works in this regard. The reflection was taken yet one step further with an example of how the provided lines of thought can be put into practice. An improved version of STAVIZ was designed using rapid paper sketching and was presented with the hope to inspire future work.

Overall, the contributions of this thesis are the following :

- A user-oriented taxonomy allowing to filter relevant existing text visualization techniques
- An overview of the previous research on relationships visualization
- A software helping analysts to perform structural analysis which received positive feedback from the participants of the user evaluation and the members of the EFFaTA-MeM research project
- An evaluation of STAVIZ based on a proposed evaluation grid that provided relevant lines of thought on how to make STAVIZ more helpful to structural analysts. The grid can also be used as a list of guidelines for anyone undertaking the development of a tool for structural analysts
- A concrete example on how these lines of thought can be used to propose an improved version of STAVIZ

However, the methodology used to answer the second research question is not immune to validity questioning.
Firstly, the six criteria of the grid were deducted based on few references, namely (Wallemacq et Jacques, 2001) and (Wallemacq *et al.*, 2004). A more extensive research in this regard could have provided additional insights, and consequently additional criteria for the grid. It can thus safely be assumed that the evaluation grid is not fully complete.
Secondly, only one structural analysis expert was involved in the evaluation process. Furthermore, the expert had not used all the evaluated techniques beforehand. An evaluation involving more structural analysis experts would have provided more extensive and insightful results. These would in turn have laid the foundation of more improvement lines of though.

Despite these methodology issues, the proposed grid proved itself able to provide insightful results when applied to a concrete case. Those results allowed designing an improved version of STAVIZ that is simpler to use and hopefully substantially more useful to structural analysts than the current implementation of STAVIZ.

The gaps in the methodology and the provided lines of though open the way for promising future work in the field of visualizations for structural analysis. For example, a similar evaluation grid could be constructed with or by several structural analysis experts and compared to the one proposed here. Another example of future work would be reviewing other existing techniques with the evaluation grid to gather more insight and propose additional lines of thought. Another different approach would be to design a novel visualization tool from scratch with structural analysts.
These examples only scratch the surface of possible future work. The fact that there is always room for someone to bring new ideas or to question previous results is what makes the beauty of the research in this field.

"I would rather have questions that can't be answered than answers that can't be questioned." (Richard Feynman)

Bibliography

Alencar, A. B., de Oliveira, M. C. F. et Paulovich, F. V. (2012). Seeing beyond reading: a survey on visual text analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):476–492.

Amar, R., Eagan, J. et Stasko, J. (2005). Low-level components of analytic activity in information visualization. *In Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 111–117. IEEE.

Bostock, M., Ogievetsky, V. et Heer, J. (2011). D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309.

Cao, N. et Cui, W. (2016). Overview of text visualization techniques. *In Introduction to Text Visualization*, pages 11–40. Springer.

Chi, E. H.-h. (2000). A taxonomy of visualization techniques using the data state reference model. *In Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 69–75. IEEE.

Collins, C., Carpendale, S. et Penn, G. (2009). Docuburst: Visualizing document content using language structure. *In Computer graphics forum*, volume 28, pages 1039–1046. Wiley Online Library.

Critchfield, T. S. (2017). Visuwords®: a handy online tool for estimating what nonexperts may think when hearing behavior analysis jargon. *Behavior Analysis in Practice*, pages 1–5.

Culy, C. et Lyding, V. (2009). Visualization of linguistic information. University lecture retrieved from `https://weblicht.sfs.uni-tuebingen.de/webservices/culy\_lyding\_weblicht\_visualization\_how\_to.pdf`.

Diakopoulos, N., Elgesem, D., Salway, A., Zhang, A. et Hofland, K. (2015). Compare clouds: Visualizing text corpora to compare media frames. *In Proc. of IUI Workshop on Visual Text Analytics.*

Gan, Q., Zhu, M., Li, M., Liang, T., Cao, Y. et Zhou, B. (2014). Document visualization: an overview of current research. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(1):19–36.

Ghoniem, M., Fekete, J.-D. et Castagliola, P. (2005). On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135.

Gibson, H., Faith, J. et Vickers, P. (2013). A survey of two-dimensional graph layout techniques for information visualisation. *Information visualization*, 12(3-4):324–357.

Graneheim, U. H. et Lundman, B. (2004). Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness. *Nurse education today*, 24(2):105–112.

Hearst, M. A. (1995). Tilebars: visualization of term distribution information in full text information access. *In Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 59–66. ACM Press/Addison-Wesley Publishing Co.

Hertog, J. K. (2010). Quantitative text analysis. University lecture retrieved from `http://www.uky.edu/CommInfoStudies/JAT/Telecommunications/hertog/TEL\_300/Presentations/`.

Holten, D. et Van Wijk, J. J. (2009). Force-directed edge bundling for graph visualization. *In Computer graphics forum*, volume 28, pages 983–990. Wiley Online Library.

Jalali, A. (2016). Supporting social network analysis using chord diagram in process mining. *In International Conference on Business Informatics Research*, pages 16–32. Springer.

Jansen, Y. et Dragicevic, P. (2013). An interaction model for visualizations beyond the desktop. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2396–2405.

Kamada, T. et Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15.

Keim, D. A. (2002). Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8.

Koch, S., John, M., Wörner, M., Müller, A. et Ertl, T. (2014). Varifocalreader—in-depth visual analysis of large text documents. *IEEE transactions on visualization and computer graphics*, 20(12):1723–1732.

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S. J. et Marra, M. A. (2009). Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645.

Kucher, K. et Kerren, A. (2015). Text visualization techniques: Taxonomy, visual survey, and community insights. *In Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pages 117–121. IEEE.

Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D. et Henry, N. (2006). Task taxonomy for graph visualization. *In Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–5. ACM.

Lee, B., Riche, N. H., Karlson, A. K. et Carpendale, S. (2010). Sparkclouds: Visualizing trends in tag clouds. *IEEE transactions on visualization and computer graphics*, 16(6):1182–1189.

LIU, S., CUI, W., WU, Y. et LIU, M. (2014). A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393.

MANGUIN, J.-L. (2005). La dictionnairique internet: l'exemple du dictionnaire des synonymes du crisco. *Corela. Cognition, représentation, langage*, (HS-1).

MILGRAM, S. (1976). Psychological maps of paris. *Environmental psychology: People and their physical settings*, pages 104–124.

MILLER, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

PALEY, W. B. (2002). Textarc: Showing word frequency and distribution in text. *In Poster presented at IEEE Symposium on Information Visualization*, volume 2002.

PANCHENKO, A., ADEYKIN, S., ROMANOV, A. et ROMANOV, P. (2012). Extraction of semantic relations between concepts with knn algorithms on wikipedia. *In Concept Discovery in Unstructured Data Workshop (CDUD) of International Conference On Formal Concept Analysis, Belgium*, pages 78–88.

PIRET, A., NIZET, J. et BOURGEOIS, E. (1996). *L'analyse structurale: une méthode d'analyse de contenu pour les sciences humaines.* De Boeck Supérieur.

PURCHASE, H. C. (2000). Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with computers*, 13(2):147–162.

RUSU, D., FORTUNA, B., MLADENIC, D., GROBELNIK, M. et SIPOŠ, R. (2009). Document visualization based on semantic graphs. *In Information Visualisation, 2009 13th International Conference*, pages 292–297. IEEE.

SAINT-GEORGES, D. (2004). Note de synthèse concernant evoq: un logiciel d'analyse et de visualisation de données textuelles. rapport final d'activité à destination de la fondation francqui. Namur : 2004, 22p.

SHNEIDERMAN, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *In Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE.

ŠILIĆ, A. et BAŠIĆ, B. D. (2010). Visualization of text streams: A survey. *In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 31–43. Springer.

TAN, A.-H. *et al.* (1999). Text mining: The state of the art and the challenges. *In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, volume 8, pages 65–70.

VAN HAM, F., WATTENBERG, M. et VIÉGAS, F. B. (2009). Mapping text with phrase nets. *IEEE transactions on visualization and computer graphics*, 15(6).

Viegas, F. B., Wattenberg, M. et Feinberg, J. (2009). Participatory visualization with wordle. *IEEE transactions on visualization and computer graphics*, 15(6).

Wallemacq, A. et Jacques, J.-M. (2001). Semantic landscapes. Paper presented at the 17th Egos Conference Stanting Working Group on the Philosophy of Organization, Lyon.

Wallemacq, A., Jacques, J.-M. et Bruyninckx, V. (2004). *Dans le sillage des mots...: EVOQ. Logiciel de cartographie cognitive.* Presses universitaires de Namur.

Wattenberg, M. (2002). Arc diagrams: Visualizing structure in strings. *In Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 110–116. IEEE.

Wattenberg, M. et Viégas, F. B. (2008). The word tree, an interactive visual concordance. *IEEE transactions on visualization and computer graphics*, 14(6).

Yi, J. S., ah Kang, Y. et Stasko, J. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231.