ConTEXTual Scripts: A visual exploration of scripts in their context

Tamar Rucham

A Thesis in the Field of Information Technology

for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

May 2014

Abstract

The visual shapes of characters today stem from a long history of evolution and is deeply rooted in the historical context in which the scripts evolved. We wish to enable exploration of the visual themes of scripts in their historical and geographical context. In order to create a platform for comparison between scripts we have analyzed the visual data of 11 different scripts. Calculating on an aggregation of lines and curves to create average glyph information for each script, we have defined a distance method to measure visual similarity. To validate our results we have randomly split each character set in two and measured the calculated distance between the parts. Combined with historical and geographical information, we have used this data to create an interactive web-based visualization. The visualization allows users to assess the similarity of different scripts and explore the contextual information that may be leading to this relative similarity or disparity. It provides insight into the evolution of the scripts, their geographic context and lays out some of the visual themes these scripts posses. Very little research is done to the best of our knowledge in the exploration of multiple scripts from a visual perspective. We hope this work can form a platform to build future research in this field.

Acknowledgments

I would like to thank my advisors professor Hanspeter Pfister and Alexander Lex who inspired, encouraged and guided me through this thesis. It is in professor Hanspeter Pfister course on visualization that many of the ideas developed, and without the guidance, thoughts and suggestions of Alexander Lex this thesis would not have been what it is.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

The focus of this thesis is to explore visual themes of different scripts. We use the word script to refer to the character set used by different writing systems to encode spoken language. For example, the Latin alphabet is a script used by many different writing systems to express languages such as English, German, French and Malay. In the field of linguistics some research exists on the development of specific scripts as discussed this in Chapter 2. However the goal of this project is not to explore the evolution from past to present, but rather to compare different current scripts in light of their historical connections.

1.1 Background

The visual aspect of written language played an important role in the history of writing. Before written language as we know it today was *proto–writing*, a picture writing system. Proto–writing uses ideograms or pictograms - graphic symbols that represent ideas or objects respectively - and does not directly translate to specific words from spoken language. The shape of the symbols conveyed the information and a reader would not necessarily need to know the spoken language of the writer in order to gather the information contained in the symbols. When most *true-writing* evolved (true-writing being a system in which the entire content of spoken language can be encoded), symbols that represent

whole words were used to encode information (logograms). Again the shape of the symbol was related to the information, but unlike in picture-writing systems, they represented specific words of the writer's language. From there, the phonetic system stemmed, in which symbols represent sounds that are combined to phonetically construct words from spoken language. The shape of the symbol no longer has a specific meaning; rather it can be combined to create many different words. Some scripts, such as the Chinese character set or Egyptian hieroglyphs, preserved both systems such that symbols can function both as logograms and as phonemes. In general, most scripts lost the connection that used to exist between the visual shape and the meaning. However the visual aspect of the scripts we use today stemmed from a long history of evolution. (XXX add references)

1.2 Motivation and Task Definition

The visual themes of scripts stem from a long history of human culture and are a fascinating field from visual and historical perspectives. While there is various in depth research both in the fields of design, data visualization and linguistics regarding different scripts, very few resources exist that aim to explore and compare different scripts concurrently. We feel this poses a new area to perform innovative research and expose a new perspective on a tool we use daily – the characters we use to encode information.

The goal of this project is to expose the themes of different scripts and allow a visual exploration of them as a first step. Second to enable an exploration of historical and geographical data in light of their visual connection. We aim to excite curiosity in the users wonder and explore scripts, both familiar and unfamiliar to them. We hope this can be a platform to inspire a new perspective and invite more research on the topic.

1.3 Result

The end result of this project is a web based interactive visualization that compares 11 “live” scripts (scripts that are in use today), all of which stemmed from the Egyptian Hieroglyph origin. The visualization contains several parts that are linked to each other. On the left there is an evolutionary tree of the scripts leading up to a heatmap, which maps the similarity between all scripts. An information area that displays specific information when comparing specific scripts is to the right (see Figure 1).

In the heatmap, color intensity reflects the similarity score calculated by our algorithm – light colored rectangles represent two scripts that are less similar than those with a stronger shade. The heatmap diagonal is the intersection of scripts with themselves (identity) and since the similarity of a script to itself is always 1, it contains no added information. Therefore we chose to overlay these rectangles and use this space in order to show a small scatter plot of the letters distribution for the corresponding script. This scatter plot shows the distribution of

the letters within the script, with respect to the number of lines and curves. All scripts are identically scaled so this small multiple can be used to compare scripts by their letters distribution.

Selecting one of the rectangles representing the similarity between two scripts highlights the two scripts and displays comparison information on the right. The comparison section shows representative characters, an enlarged version of the scatter plot with the character set as the markers and a world map with current script distribution. It also updates the legend at the bottom of the heatmap to display the current similarity score. The name of the script at the top of the information section is a link to its Wikipedia article to allow further investigation. Each of the two compared scripts is represented by a distinct color throughout the visualization; countries that use both scripts are in a darker shade on the map as seeing in Figure 1.

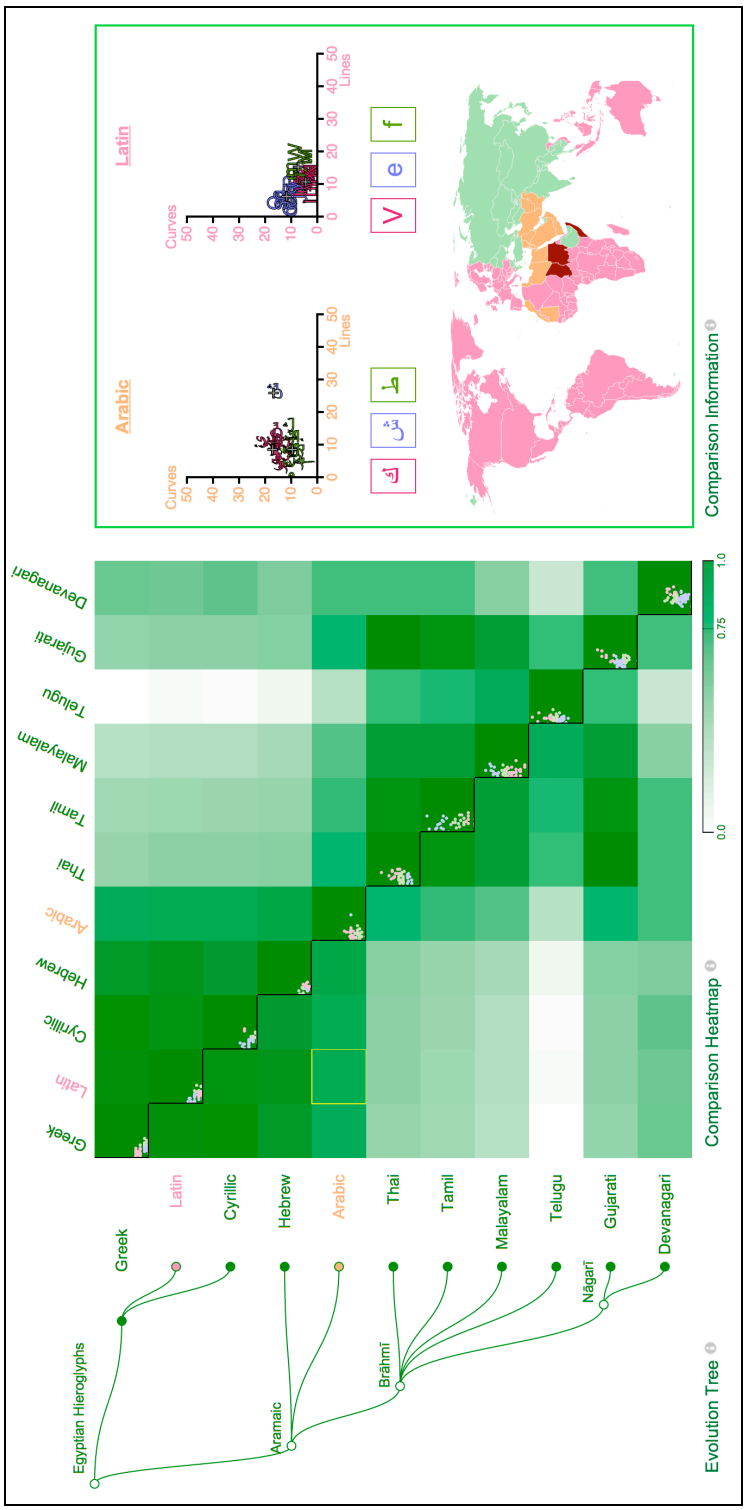

Figure 1: The full visualization with the comparison between Arabic and Latin selected.

1.4 Organization of this document

The related work in chapter 2 covers work done both in the field of linguistics and in the field of data visualization. Chapter 3 contains the implementation details relating technical details of the project. Chapter 4 describes the methods used to implement this project and the reasoning behind them. Chapter 5 is conclusions and summery.

Chapter 2 – Related Work

In this section we will review related work, from the domains of linguistics, visualizations and art. We will survey existing visualizations on scripts, visualizations on languages (evolution etc.) and text visualizations related to writing systems as well as typography and some art projects that are tangent to this work. We also review heatmaps, which is the main visualization technic used in this project. Lastly we will present previous projects, which formed the groundwork for this project.

2.1 Scripts visualizations

In the field of linguistics some visualizations have been made to illustrate the evolution of different scripts. These visualizations, while they provide insight into the historical context of the scripts, commonly do not investigate and discuss the visual themes of the scripts. Moreover, these visualizations tend to provide a depth exploration, from past to present, and be less focused on a breadth view to compare parallel scripts.

Fradkin, for his class on the History of Alphabets at the University of Maryland, has made several animations (executed by Charlie Seljos) that demonstrate the evolution of different alphabets, for example the evolution of the

Latin alphabet from Phoenician displayed in Figure 2. It visually shows how letters gain their visual form and displays the change of style over the course of history. It provides an interesting insight into how the characters transformed into what we know and use today. While it is a fascinating exploration of the history and evolution it offers no analysis of main visual themes, and although the evolution of several alphabets is available, the focus is in exploring each script path separately, making it difficult to compare the visual themes of different modern scripts (Fradkin, 2000).

Ada Yardeni, in her Hebrew book HarpatkaOT, lays down the transformation of each character in the Hebrew alphabet from the proto-Sinaitic origin to their modern form. The book also visually displays the parallel path to the "sibling" characters in Latin and Arabic scripts, which stemmed from the same origin (see Figure 3 for an example letter evolution). Thus this book allows an in depth, per character, analysis of these scripts (Yardeni, 1993).

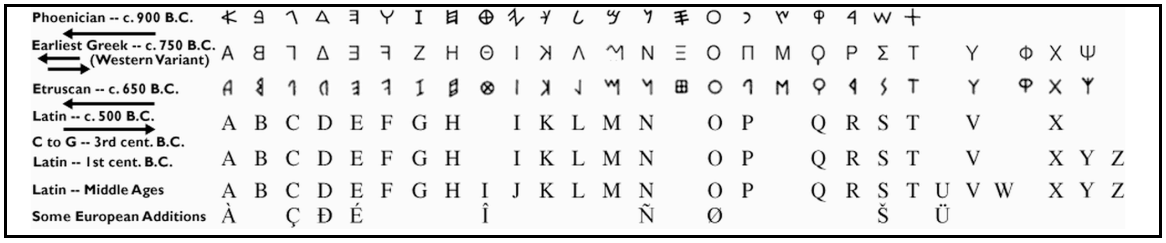

Figure 2: The result of the animation done by Fradkin on the evolution of the Latin alphabet from its Phoenician origin.

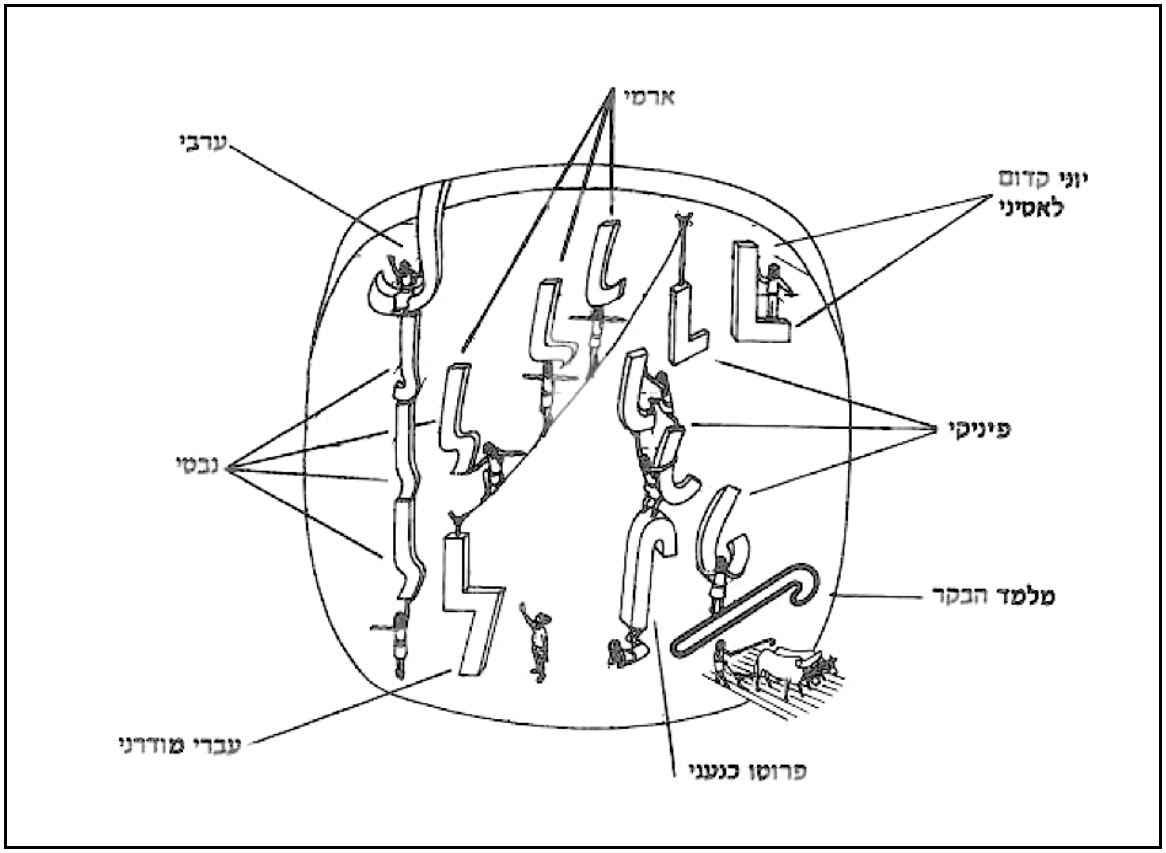

Figure 3: The letter L, ל and ل in Latin, Hebrew and Arabic respectively derived from their proto-Sinaitic origin, as illustrated in Yardeni's book.

2.2 Linguistics and Language Visualizations

In this subsection we will review visualizations that illustrate relationships between languages and writing systems. The datasets that are visualized in these works are related and we consider it tangent to the data presented and visualized in this research. They attempt to represent their data in the historical

and geographical context and sometimes use methods similar to the ones used in this thesis.

An article by Yasin in Stanford University[1] describing Gell-Mann & Ruhlen research on sentence structure relation and development between spoken languages, uses visualization technics similar to our. The said article uses a tree to express the evolution of languages with color-coding to indicate the sentence structure used in each. A world map is also displayed to allow for the comparison to geographical data Figure 4. However the color-coding used in the map does not correlate to the color encoding of the tree, making the visual comparison challenging (Gell-Mann & Ruhlen, 2011).

Teresa Elms created a visualization of the lexical distances between European languages[2] based on linguistics research by Tyshchenko (Tyshchenko, 2000). The color and size is used in the visualization to relate historical information about those languages, while the distance and lines are

[1] http://humanexperience.stanford.edu/languagetree

[2] http://elms.wordpress.com/2008/03/04/lexical-distance-among-languages-of-europe/

used to relate the lexical distance between them. This allows the user to visually compare the relation, historically and lexically between the languages (see Figure 5).

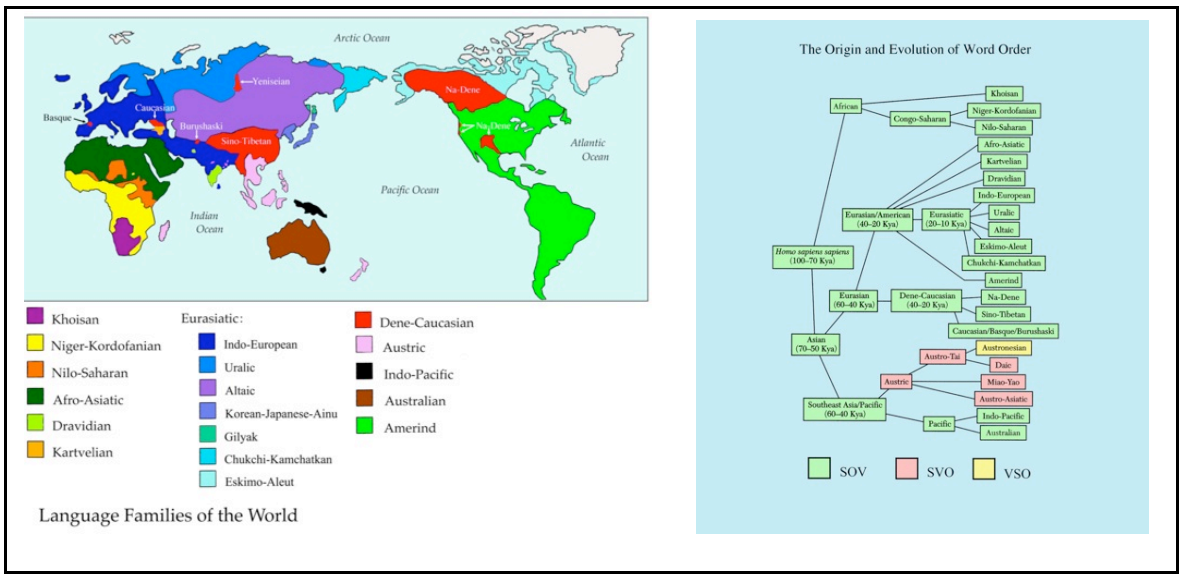

Figure 4: The color coded map and the tree representing sentence structure relation and evolution based on Gell-Mann paper.

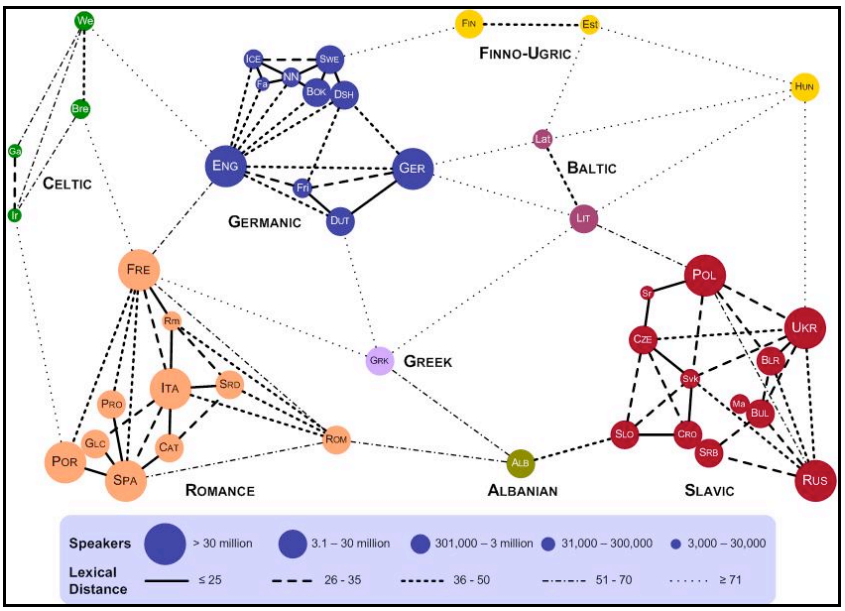

Figure 5: A visualization by Teresa Elms mapping the lexical difference between European languages.

2.3 Writing Systems Text Visualization

There is a significant body of work on visualizing text in many different ways, here we will present some that explore the text with an emphesis on the context of writing systems and language structure.

Abdul-Rahman et al created a tool that analyses text in general and poetry in specific in a visual way (see Figure 6). The text is analyzed both phonetically and semantically including the connections and features of the different blocks. The approach also uses a visual tool in order to analyze text, and is an example of collaboration between two remote domains in order to create a new tool for analyzing poetry. However this tool is used to investigate the inner structure of

the text rather than it's visual form and does not currently promote the comparison between different scripts of even different text segments (Abdul-Rahman, et al., 2013).

Collins et al in their project DocBurst visualize text based on language structure. They extend the basic idea of visualizing word frequencies in text and give the lexical language context. The user selects a word and based on their database the visualization is populated with all of its hyponyms, highlighting the ones that occur in text by frequency (see Figure 7). This provides an interesting insight into text from a lexical perspective (Collins, Carpendale, & Penn, 2009).

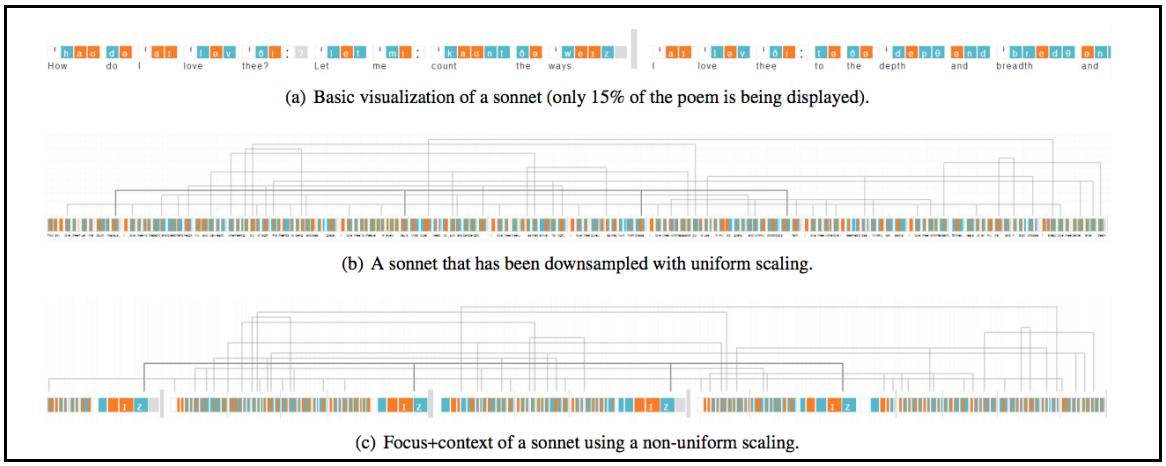

Figure 6: An image from Abdul-Rahman et al rule-based visual mappings showing three different layouts ofa poem.

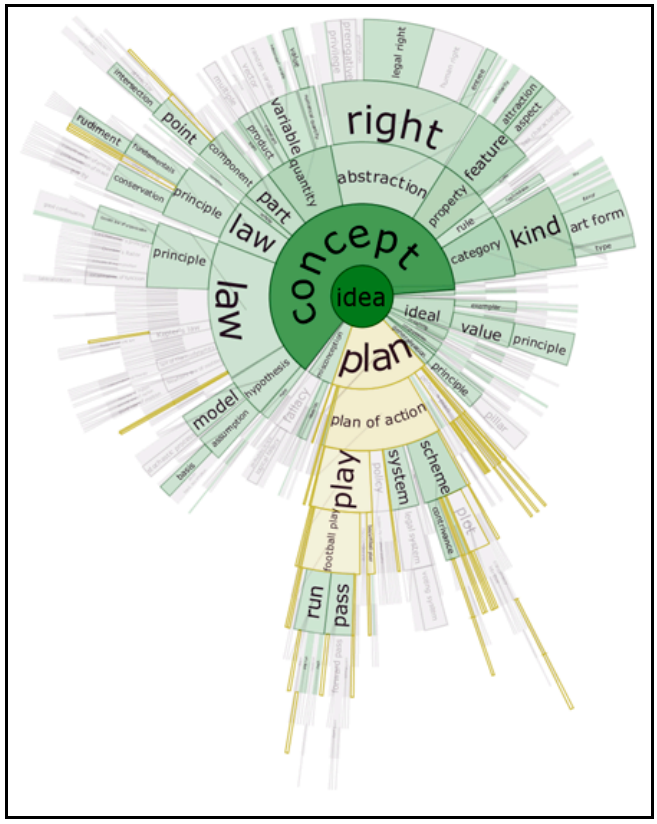

Figure 7: A DocBurst analysis of a science text book rooted at the word 'idea' with a search on 'pl'.

2.4 Typography and Font Design

Today the evolution of letters and scripts has somewhat changed due to the persistent nature of print and computers. Even when a need comes to create a new character, such as in the transformation of Khoisan into written language, existing glyphs are used. Khoisan is the language spoken by the San people of southern Africa, and until recently it was only a spoken language, without a writing system to encode it. In recent efforts by linguists and local people, a

writing system was created using a Latin derived alphabet. However, their language contains click sounds that are unique and cannot be represented with the common characters. Therefore glyphs such as the pound (#) and exclamation (!) marks are used to represent those sounds (XXX mark name and reference).

Yet within those limitations on the evolution of glyphs and scripts, there is significant effort and room for innovation in the design of fonts. Many designers and marketers spend significant thought and effort into font design. Bringhurst in his renowned (XXX) book describes the art and importance of font design. He also lays out an interesting historical overview of changes in the Latin alphabet fonts and how they reflect cultural changes of the time, see chapter XXX there (Bringhurst, 2004).

Abou Rjeily in her book tries to bridge cultural gaps using an exploration of Latin and Arabic character sets. She uses visual means to introduce Middle Eastern culture as shown in Figure 8. Her attempt at mixing visual themes from Arabic and Latin characters is a fascinating attempt at exploring visual differences and their historical and cultural roots (Abou Rjeily, 2011).

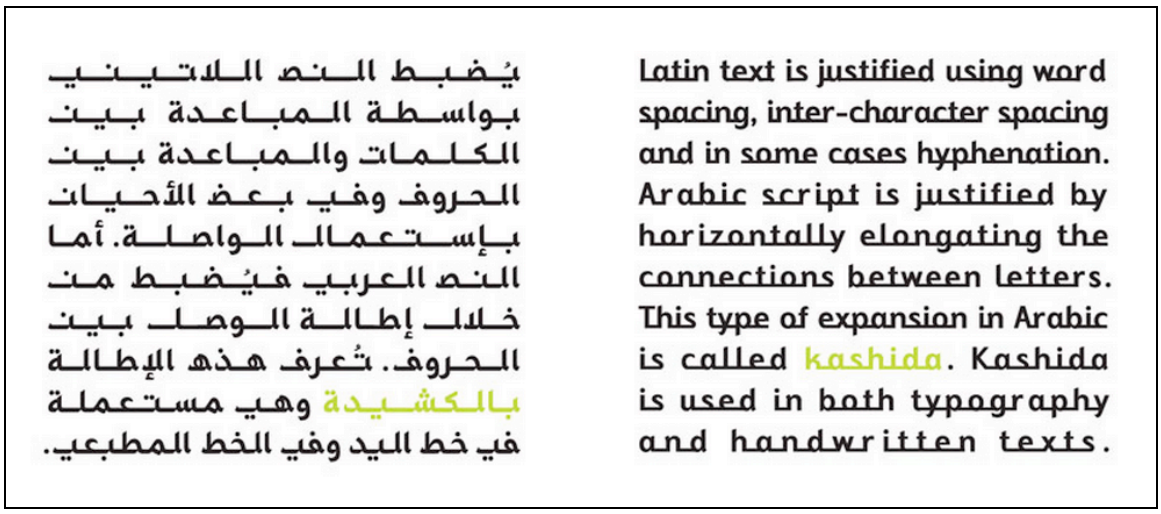

Figure 8: An attempt by Abou Rjeily to visually bridge the cultural gap by creating a Latin font that is inspired by Arabic visual themes.

2.5 Art Related Works

Wordcollider[3] is an artistic visualization done by Moritz Heller inspired by particles collision that accelerates two phrases into each other, giving a different visual theme to each letter's phonetic characteristics (Figure 9). The end result is

[3] http://vimeo.com/37015401

a visual representation of the two phrases phonetically. Though the approach is interesting and creates an intriguing visual “footprint” of the text, it does not allow for exploration or comparison and is in essence an attempt to visualize sounds (phonetic characteristics). Moreover this approach “visualizes” phonetics, in other words, attempts to give a visual form to sound, while this thesis focuses on analyzing the character sets themselves (Heller, 2012).

Null sets is an artwork that visualizes text files as images, representing their size and structure (see Figure 10). The result is an abstract image of varying color that creates a visual rhythm that is based on the text and therefore represents the structure. Although this project takes a similar approach in the concept of visualizing text and despite the fact is allows for a comparison between different text segments, the end result is not derived by and does not indicate of the visual themes of the original text (Szczepanski & Meaney).

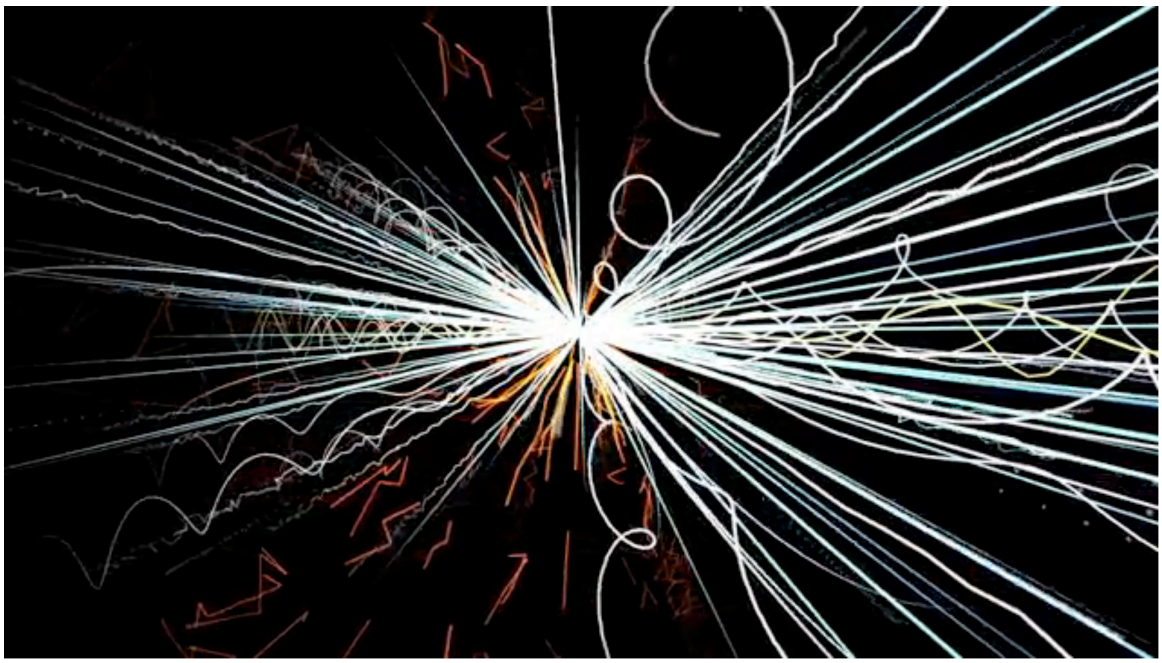

Figure 9: An image out of the wordcollider video.

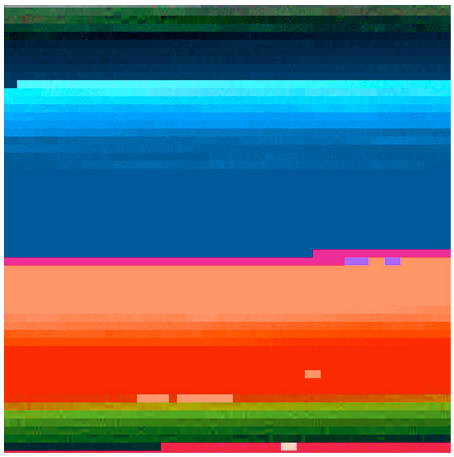

Figure 10: The Wikipedia article about JPEG converted into an image using the null sets application for representing text in an image.

2.6 Heatmaps

The main visualization technic used in this thesis is the heatmap; Here we will briefly survey this and related comparable technics.

Wilkinson and Friendly describe the history of heat map matrices while describing their most common use cases and applications (Wilkinson & Friendly, 2008). Rufiange et al suggests a new method of combining trees and heatmap matrices when the data contains multiple dimensions by combining color-coding and other methods of distinguishing the hierarchical data. Figure 11 provides an example of their suggestion (Rufiange, McGuffin, & Fuhrman, 2012). While we do use a combined tree and heat map, this approach is more appropriate when dealing with more complex datasets.

Ghoniem et al compare using a heatmap to a linked node when representing rich data. They show that when presenting large and complex data sets, a heat map matrix representation can outperform in some tasks of conveying data to users compared to a linked node (Ghoniem , Fekete , & Castagliola, 2005). While our data set is small and can be represented as a connected graph it does not intuitively represent a network. Also the heatmap representation selected allows an easier exploration of the data and the possibility of further expanding the data set without compromising the readability and accessibility of the visualization. This is further discussed in Section 4.8. Seo and Shneiderman describe the importance of clustering in reading heatmap

displays and suggest a new method of presenting multidimensional heatmaps (Seo & Shneiderman , 2002).

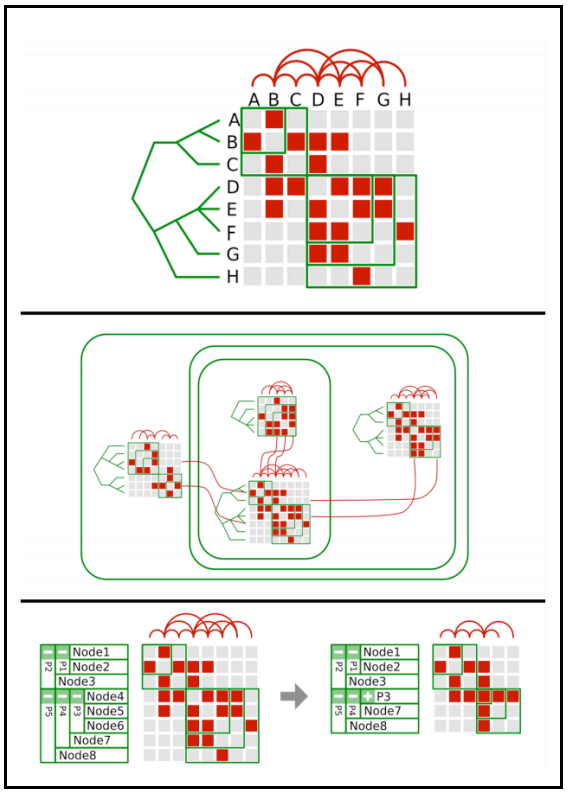

Figure 11: An image describing Rufiange et al suggestion for combining tree and matrix displays into one. The different color coding in the heat map itself are to identify the tree (green) hirarchy.

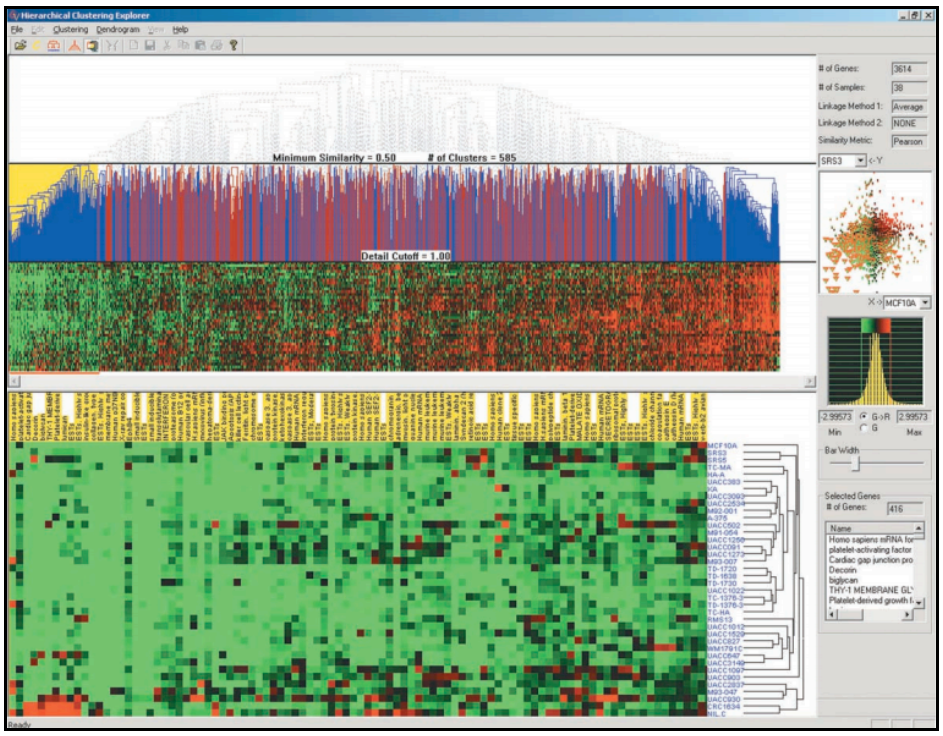

Figure 12: Seo and Shneiderman's Hierarchical Clustering Explorer's compressed overview.

2.7 Previous projects

This section describes two projects, *letters space distribution* and *the origin of languages and their scripts*. These projects were done as part of the visualization class with professor Hanspeter Pfister and together they form the groundwork that led to this thesis project.

Letters space distribution is a project that aims to explore the distribution in space of glyphs in text segments of varying writing systems. It allows for the exploration of the space distribution for a writing system used by a specific

system and a comparison between the different writing systems. The space distribution is obtained based on text segments, which provide sets of characters from each writing system. The bitmap, the image of the character as a 2D array of pixels, is retrieved for each character. Then the bitmaps of all the characters are overlayed, counting the number of times each pixel is visited. This generates a gray scale “heat map” showing how often a pixel is used by the characters in the given text segment and provides a visual insight into how the characters of that writing system are distributed in space. Note that all characters in given text segments were used, therefore if a character repeated multiple times it had a stronger impact on the resulting space distribution. For example Hindi’s space distribution (Figure 13) shows almost all characters have a line across the top, since these pixels are most strongly colored.

The first part of this visualization (Figure 14) contains a small multiple showing the space distribution derived from 10 different languages, on the first row - English, Portuguese, German, French and Malay which all use a latin derived alphabet; on the second row - Hebrew, Arabic, Hindi, Thai and Chinese which each use their own distinct writing system.

Selecting a specific writing system brings up a page that displays a larger interactive version of the space distribution for that system on the left, all the characters that appear in the text as a bubble chart on the right, and the original text at the bottom of the page (Figure 15). The size of the bubble corresponds to the number of time that character appears in the text, which is displayed at the

bottom. Hovering over a pixel in the space distribution section on the left highlights that pixel in orange as well as the corresponding letters (i.e. letters that occupy the highlighted pixel) in the bubble chart on the right (Figure 15).

Since the space distribution is based on the frequency and use of specific characters, we expected to find a greater difference between the writing systems that use a Latin derived alphabet, however this visualization shows that the space distribution of these systems is remarkably similar. The Malay text is slightly different as it seems to use the letter *a* more heavily than *e*, unlike the other latin derived writing systems. This slight difference as well as the high similarity of the other latin derived scripts may be explained by the fact that the spoken languages English, Portuguese, German and French all have a common Indo-European origin and share a closer history than Malay. Chinese (Figure 16) was probably the most interesting to note as it was the most distinctly different - it showed very dense characters with even distribution over a square, and a significantly higher amount of characters used with a smaller occurrences rate.

Figure 13: Hindi's space distribution.

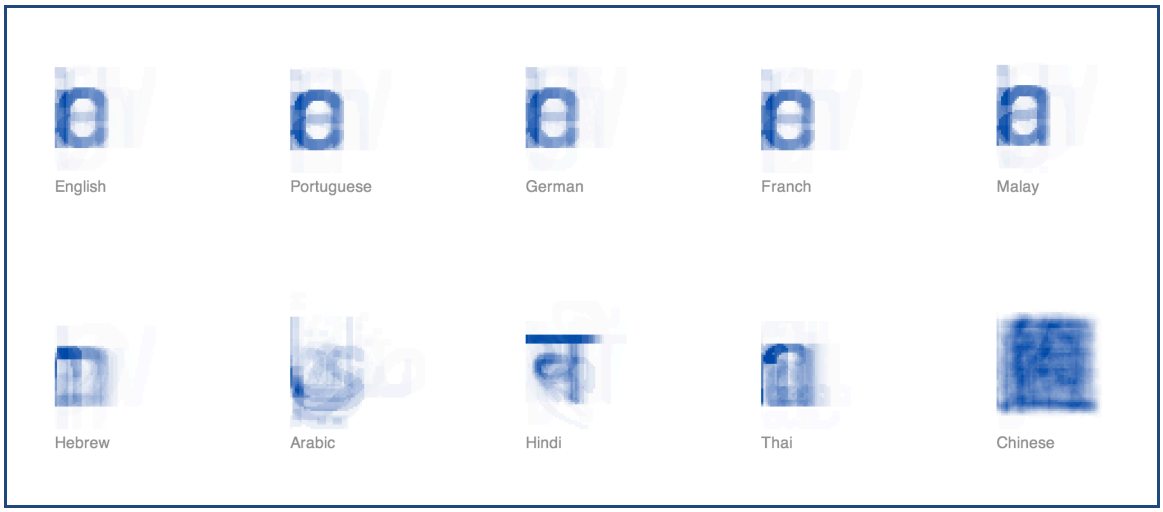

Figure 14: Space distributions of 10 selected writing systems.

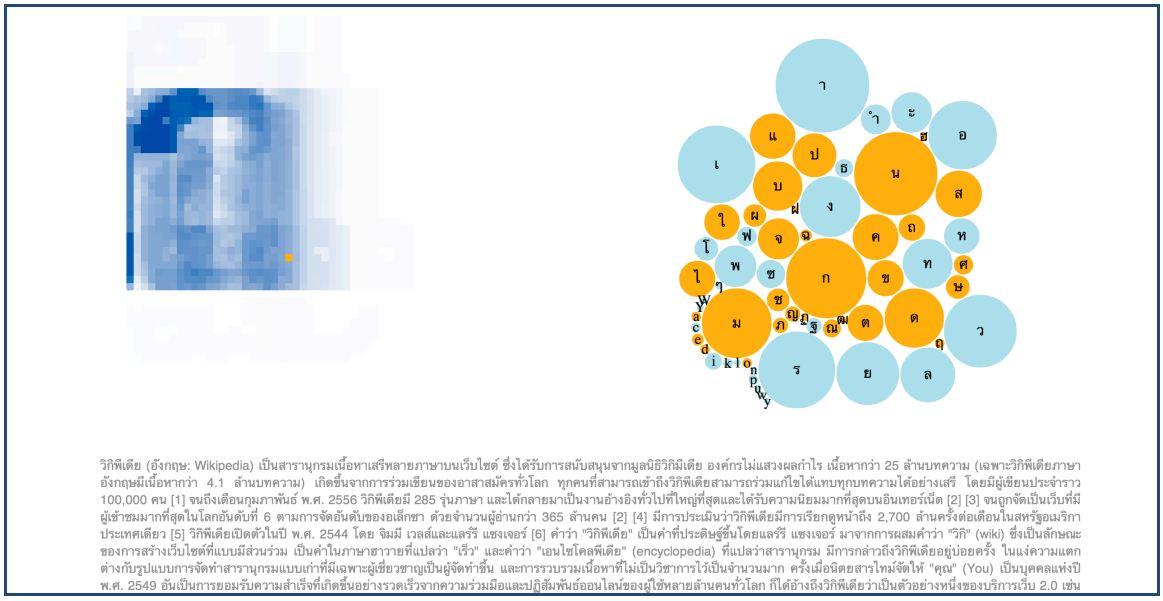

Figure 15: The space distribution and characters of Thai with a specific pixel and corresponding letters highlighted.

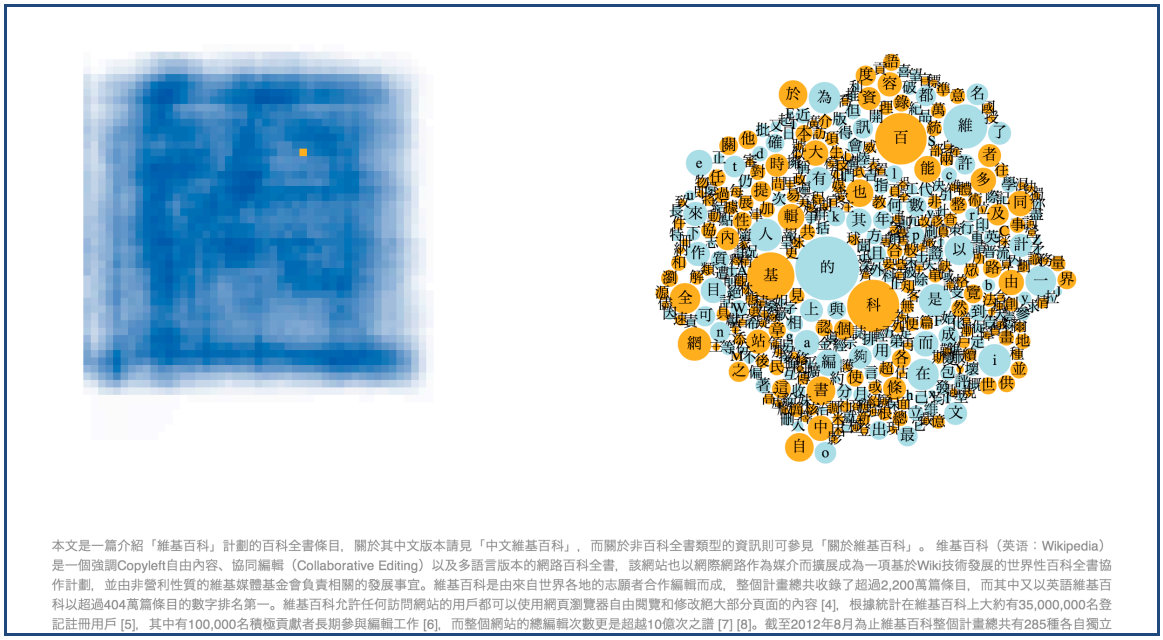

Figure 16: The space distribution and characters of Chinese with a specific pixel and corresponding letters highlighted.

The origin of languages and their scripts is a project that aims to collect and visualize the evolution tree of different languages and their writing systems. Origins of spoken languages are displayed on the left and origins of writing systems are displayed on the right. Clicking on a language family brings up the tree for that family (Figure 17). Languages that have an alphabet connected to it are larger and in full color. Selecting one of those nodes displays the alphabet node at the same level, connected to it's origin. The selected node is highlighted while the rest are grayed out (Figure 18). At the bottom there are details on demand - a frame displaying the Wikipedia article about the language or the alphabet (Figure 19). Selecting the alphabet will open the writing system tree,

with that node already selected. When a script tree is open, selecting one of the selectable scripts will present a list of all languages that use that alphabet.

Figure 17: The Turkic origin evolution tree.

Figure 18: The language Bashkir is selected and the corresponding alphabet is displayed.

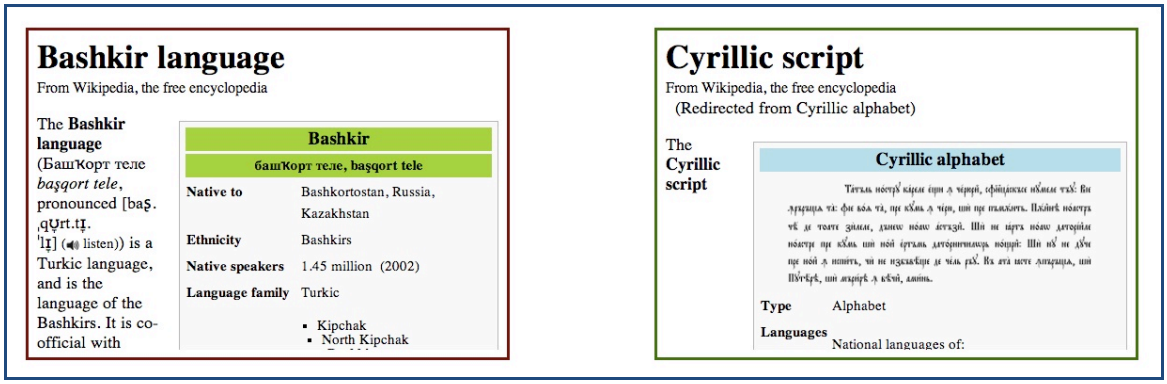

Figure 19: Additional information from Wikipedia on the selected language and script.

Chapter 3 – Implementation

The chapter describes the implementation details from a technical perspective. The project has two implementation parts – data gathering and analysis and web-based visualization.

3.1 Data Collection and Analysis

The data was collected and analyzed using python with the help of the Numpy library for matrix manipulation and the FreeType_py library for glyph information retrieval. The FreeType_py library uses ttf files (true type font) to provide character information. Many operating systems today come with the most common fonts already installed, however many others can be downloaded. Jack

Kilmon[4] and Ecological Linguistics[5] both offer font files for ancient scripts. This opens the possibility of a visual exploration of scripts that are no longer in use as a possible continuation of this project.

Using a specific ttf file and Unicode indices per scripts, the code analyzes the characters and outputs the general data to a json file. Secondary processing files then use this output in two ways. One code path generates the required information for the heatmap – calculating and storing the similarity score between every combination of scripts indexed properly for easy heatmap access. The other code path uses the data to generate chars relations information per script, each saved into a separate json file under the chars folder. This information is further processed to create the clusters of each script (see Figure 20).

The information for the evolution tree was scrapped from Wikipedia using the Pattern library Wikipedia API (*add reference*). The file created in code was then manually adjusted to include only the scripts that are compared in this

[4] http://www.historian.net/newindex.html

[5] http://www.lingfonts.com/

thesis. More details on scripts selection can be found in the methods section of this document.

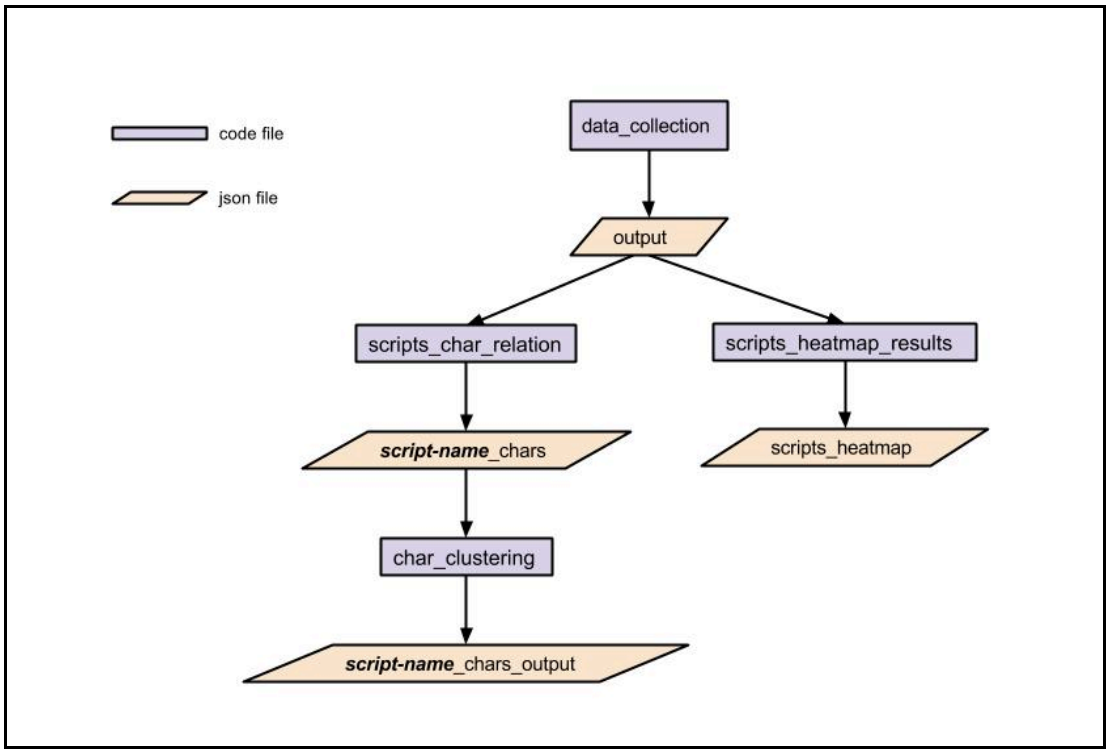

Figure 20: A diagram of the data flow in python code for data collection and analysis.

3.2 Data Visualization

The visualization is a web-based application built with html, css and javascript, using the D3 javascript library (*add reference*). The different parts of the visualization are implemented in separate files and are generated as multiple individual SVG rather than a single element in order to provide encapsulation of

each element. All these visualization parts have a javascript file that is responsible for creation and setup and a css file for individual styles.

The **tree** section implements the evolution tree. The **heatmap** section implements the main heatmap comparing the scripts along with the top labels.

The **data** section implements the comparison section on the right and includes the world map based on the topojson library (*add reference*). This section uses a static data file, which contains information such as links to their Wikipedia articles and geographic information by country codes. In addition it uses a json file containing information on country boundaries to generate and manipulate the world's map. This file[6] indexes countries by their ISO 3166-1 numeric codes. Since the list of countries per script in the static data file uses the countries ISO 3166-1 alpha-3 codes a file mapping the two was downloaded[7] and the relevant information was extracted using a simple python script.

---

[6] Coordinates downloaded from http://bl.ocks.org/mbostock/raw/4090846/world-50m.json

[7] Country information downloaded from https://github.com/mledoze/countries/blob/master/countries.json

Finally, the **char relation** section implements the per-script scatter plots (individual SVG elements), both the large version, which includes the letters, and the simple smaller version displayed inside the heatmap. This file uses the clusters of each script generated by the python script, in order to mark the clusters and the representative characters.

Chapter 4 – Methods

The section describes the methods, algorithms and approaches taken in this thesis, from data collection to analysis and finally to visualization.

4.1 Writing Systems vs. Scripts

To begin our work we had to define the scope of the project, whether we focus on writing systems, as in the project Letters Space distribution or character sets (scripts), as in The Origin of Languages and Their Scripts. A visual aspect of a writing system is far more complex than just it's script, as it also includes amongst other things, length of words and frequency of character use. While considering a comparison of the visual aspect of writing systems, we thought of using large bodies of text in different writing systems to run the analysis on. The bible was an obvious selection for its length and availability in many writing systems. This would have required some sort of validation that the length and language variety of this text is sufficient to describe the writing system. Another option is to use dictionaries, however dictionaries are not a representation of a writing system in text. Dictionaries being essentially lists of words don't seem sufficient to represent a writing system which also includes sentence structure

which effects the visual aspect of a writing script. Eventually we came to the conclusion the visual aspect of writing systems is more vague and can be broadly defined, and though it may be an interesting future project, for the scope of this thesis we will focus on scripts.

The basis for which scripts to compare was the scripts used in the previous project *The Origin of Languages and Their Scripts*, seeing we have already obtained their evolution tree in a process described later. In addition, in order to keep the scope of this thesis from getting too broad, we decided to focus on scripts that originated from the Egyptian Hieroglyphs. Scripts coming from the Oracle Bone script origin such as the Chinese character set, tend to be extremely large (over 3000 characters) and sometimes not easily defined - some writing systems use a mixture of character sets, or derived characters (*reference*).

4.2 The Free Type Library

The analysis of the scripts is done using the freetype-py library for python. The library provides both bitmap and vector information on glyphs based on a provided ttf file. We started by looking at the glyph-vector example provided with the library. Note that this example uses matplotlib to plot the glyph. The glyph information is initially divided into closed contours - closed lines, for example the letter O has two contours, the outer circle and the inner circle while the

letter S has only one contour. The closed contours are in turn divided into segments that can be either a straight line or quadratic B-splines. Each spline is equivalent to one or more quadratic Bézier curves. Each Bézier curve is defined by three outline control points – two on-curve points (A and C) and one off-curve (B). If a B-Spline curve is equivalent to more than one Bézier curve, the first and last on-curve points are provided and an implied on-curve point is interpolated between any two consecutive off-curve. The letter 'b' in Arial font with it's control points is presented in Figure 21. For our algorithm we have counted Bézier curves and not only the B-Spline curves. (XXX *Add reference)*

Figure 22 shows a few examples of characters from different writing systems with their control lines (dashed). These are single characters from a few selected scripts we have gathered in order to explore how to access and analyze the data available with this library. These are characters from the Chinese, Latin, Hebrew and Tibetan character sets (from left to right respectively). The statistics provided by the library for each character are presented in Table 1.

This experiment demonstrates the great difference in number and type of segments that can exist. There is a significant difference between the number of closed contours and segments between the Chinese and Tibetan selected characters on the one hand and Hebrew and Latin characters on the other. In addition, the Chinese and Tibetan samples differ in the type of segments as well – the Chinese sample contains many straight-line segments while the Tibetan

one has many curves. Indeed if we examine the characters ourselves, the variance in complexity and nature that is reflected in the data is clear to see.

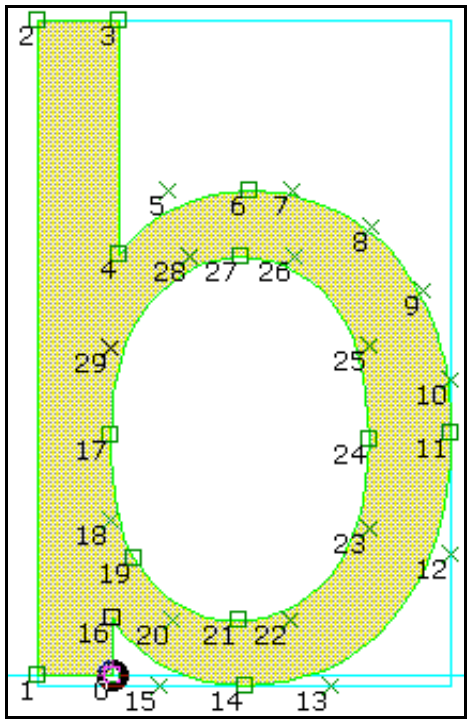

Figure 21: The letter 'b' in Arial font with control points. Point 4, 5 and 6 represent a quadratic B-spline of a single Bézier curve, where 4 and 6 are on-curve and 5 is off-curve. Points 6 to 11 are another quadratic B-spline equivalent to 4 Bézier curves with an implied point between the points 7 – 8, 8 – 9 and 9 – 10. Image from the truetype typography website[8].

[8] http://www.truetype-typography.com/ttoutln.htm

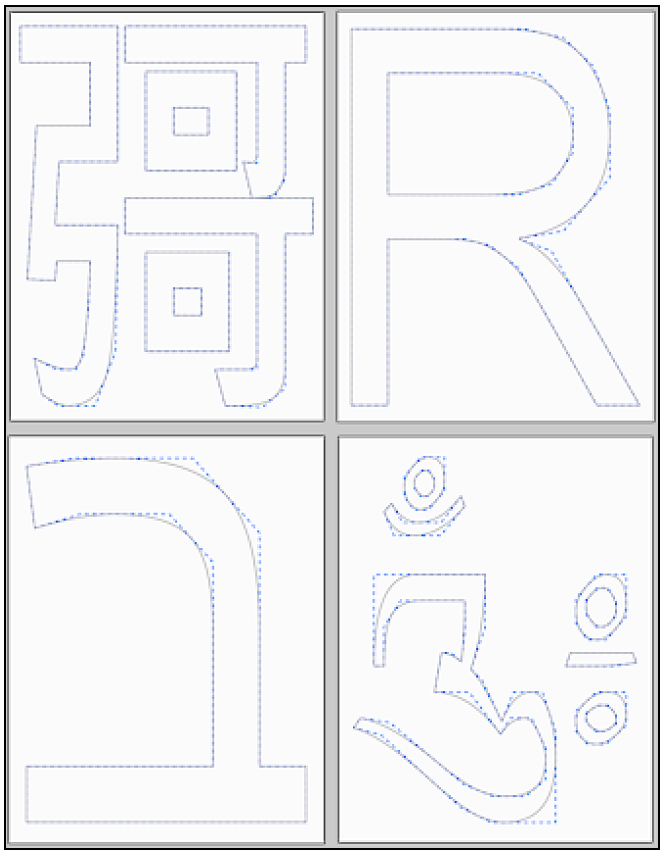

Figure 22: Sample characters from Chinese, English, Hebrew and Tibetan, with control lines of Bezier curves provided by the freetype library.

	Chinese char	English char	Hebrew char	Tibetan char
Closed contours	7	2	1	9
Total segments	57	17	12	58
Straight lines	47	11	8	9
Curves	9	6	4	49

Table 1: Data gathered on the 4 sample characters from different scripts using the freetype library. This displays number of closed contours, total segments, lines and curves for each of the character.

4.3 Defining the Dataset

Based on the information described in section 4.2, we needed to use scripts to which we have font files. However, there is great variance in visual aspect of characters between different fonts, mainly between serif and san serif fonts (XXX *what is serif / san serif*). Even within those families we wanted to choose a font that can serve as a baseline for comparison. Therefore the Arial Unicode font was selected for representing all the scripts. The Arial Unicode font covered the most scripts of the ones we have defined in the project *The origin of languages and their scripts* and more importantly this font was created, as his creator NAME described it, to be **as generic and blend as possible** (XXX reference quote). Out of the scripts we started with, the Arial Unicode ttf file supported 11 scripts – Greek, Latin, Cyrillic, Hebrew, Arabic, Thai, Tamil, Malayalam, Telugu, Gujarati, and Devanagari. Therefore we selected these scripts as our dataset to explore. For each of these scripts we manually defined the ranges of letters based on their Unicode charts (XXX *reference*).

4.4 Evolution Tree

Before diving into the glyph analysis and script comparison, we will take a moment to describe the method used to obtain the evolution tree of the scripts. This work was done as part of the project *The Origin of Languages and Their Scripts*. The codes for all the languages that have a Wikipedia article about Wikipedia were scraped using the Pattern library Wikipedia API. Then the language name was matched using an external source. The name was used to programmatically search the Wikipedia article about that language, which was in turn scraped and the linked to the language writing system was followed and scraped as well. Articles about languages and scripts have specific structure for the origins as can be seen in Figure 23. Languages that did not follow this structure were ignored. A python script, traversing the evolution list from the origin down, constructed the tree and stored the data in json files. After the generation of the data by the script, it was manually cleaned. There were some inconsistencies with trees starting points; for example some scripts ended their parent systems in the proto–sinaitic origin, though that is a child of the Egyptian hieroglyphs. Some references collected from different pages had slightly different names, such as Latin script vs. Latin alphabet. Also, to reduce the overwhelming amount of data, some of the least used languages were removed. To narrow down the displayed tree size, nodes that have a single child, a single line of connection, are not presented in the tree.

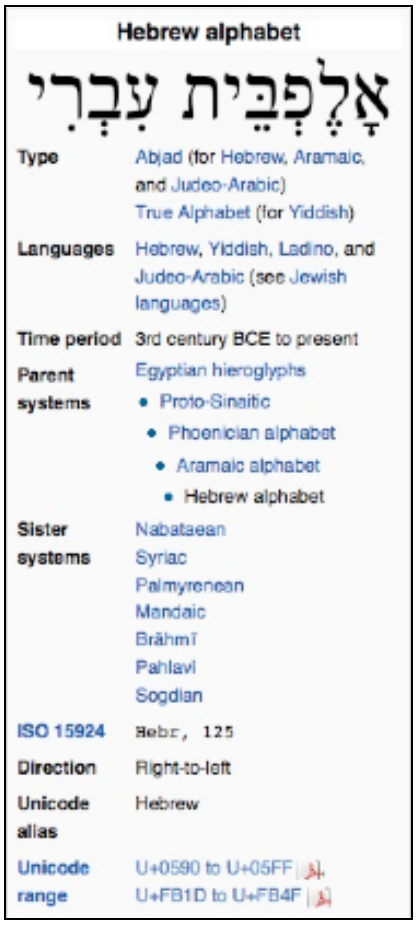

Figure 23: Details from Wikipedia for the Hebrew script showing the evolution from Egyptian Hieroglyphs to the script under the Parent systems section. This section was used to construct the evolution trees of the scripts.

4.5 Distance Formulas and Normalization

For every character in each script we calculated the number of lines and the number of curves (we also got the number of closed contours but that information is not used in this project). Based on these results we calculated the

average number of lines and the average number of curves per script. Next we needed to define a distance function, how do we measure a similarity, or disparity, between scripts. This formula is also used to measure distance between characters based on lines and curves for clustering.

The first formula we used was simply an accumulation of the differences between the number of lines and number of curves. Therefore this produces a value starting from 0 (identity) going up to the greatest difference, which for this formula was around 25 – comparing Greek and Telugu:

```
     absolute_value(script_1_lines - script_2_lines)
+ absolute_value(script_1_curves - script_2_curves)
```

Later when we came to create clusters of the letters (described in section 4.10) we started considering the number of lines and number of curves as our coordinates in a two dimensional plane. Thinking of clustering we found that a small tweak to the formula, making it a straightforward Euclidean distance formula, worked better. Besides making the clustering algorithm simpler, it gave slightly better results in the validation process described in Section 4.6. Therefore the formula that is used to calculate the similarity score is:

```
     absolute_value(script_1_lines - script_2_lines)
/ absolute_value(script_1_curves - script_2_curves)
```

We have used the same formula to calculate the similarity score and the distance between characters for clustering. For characters we used the actual number of lines and curves and for the similarity score we used the averages.

Finally we normalized the results of script similarity scores to a value from 0 (greatest disparity) to 1 (identity). This makes comparison, further calculation and representation of the data simpler and easier to read. To normalize the data we take the maximum found difference minus the calculated score, divided by the maximum. Though commonly normalization methods take the value minus the maximum, here we also wanted to reverse the direction of the results - instead of have identity at 0 and disparity at 1, we preferred to have the similarity score lower as the disparity increased. The condition preceding the formula below is of course to avoid division by 0 when the value equals the maximum data.

```
0 if value==maxData else (maxData-value)/maxData
```

4.6 Validation

In order to measure the performance and validate the algorithm described in the previous section, we created a validation script. The code randomly splits the character set in two and evaluates the similarity score. The assumption is that the two subsets of the same script should come out with a good similarity

score (0 being identity - same character set). This was done 150 times for each of the scripts and outputs the normalized average score received.

After changing the formula from the accumulation of differences into a Euclidean distance formula, we ran the validation again. The results shows that the second formula gives slightly better results with the average score of the first algorithm being ~0.909 and the second algorithm providing an average score of 0.915. The detailed results are shown in Table 2.

While finding the validation test results fairly high for such a simple algorithm it was interesting to note it provided better results with some scripts. Also, while we consider the validation results to be quite satisfactory, it still did not reach the highest similarity scores that were calculated between the scripts (see further discussion on the high similarity scores in Section 5.2). Looking into the validation process itself, we found high variance in validation results per script iteration. Meaning, while the average validation score per script was stable when calculated over 150 iterations, specific scores provided in every iteration varied greatly. Over 10 iterations of random splits of Hebrew, we found results ranging from 0.87 to 0.99. For Gujarati the range was even larger ranging from 0.57 to 0.96. Our estimate is that for such a small set (number of characters per set) the importance of the split is high. Moreover, this variance is higher for scripts with a wider distribution of characters, i.e. great differences in number of lines and curves between characters. This can be observed in the scatter plots presented in Section 4.10. Scripts that got an overall better validation tend to

have a denser character distribution. On the other hand scripts that have sparser distribution, representing greater complexity and variety of characters, such as Gujarati, present a larger range in validation based on random splits. This translates into lower validation scores to scripts with greater character distribution. Interestingly, this ties back into the similarity between scripts since there is a strong correlation between resemblances in characters distribution to the scripts similarity scores, more in Section 4.10 as mentioned.

Script	**Validation I**	**Validation II**
Telugu	0.91	0.92
Cyrillic	0.93	0.94
Greek	0.94	0.95
Malayalam	0.89	0.89
Thai	0.92	0.92
Latin	0.93	0.94
Gujarati	0.89	0.90
Hebrew	0.94	0.94
Devanagari	0.93	0.93
Arabic	0.91	0.91

Tamil	0.81	0.83

Table 2: The normalized average similarity score of each script randomly split in two over 150 iterations. A score of 1 means identity.

4.7 First Visualization as a Connected Graph

As an experiment we calculated the similarity score for 7 of scripts – Latin, Greek, Cyrillic, Hebrew, Arabic, Thai and Devanagari – and plotted them as a connected graph (Figure 24). The length of the edges, which defines the distance between the nodes, uses the similarity score as the weight. The tight group of 4 near the bottom contains Latin, Greek, Cyrillic and Hebrew, the more remote outliers are Thai and Devanagari, and Arabic is somewhere in the middle.

Curious to see the importance of the font selection, we calculated the similarity score using the Courier New font, and re-plotted as shown in Figure 25.While some elements were preserved, such as the cluster of four discussed above, we could easily see a tight cluster in the new graph that does not exist in the Arial font generated graph. This cluster consists of Thai and Devanagari; when analyzed with the Arial font these scripts have quite a difference between them, expressed as the distance between the vertices. However with analyzed with the Courier New font, the similarity score between them came out exceptionally high, which is expressed in their closeness.

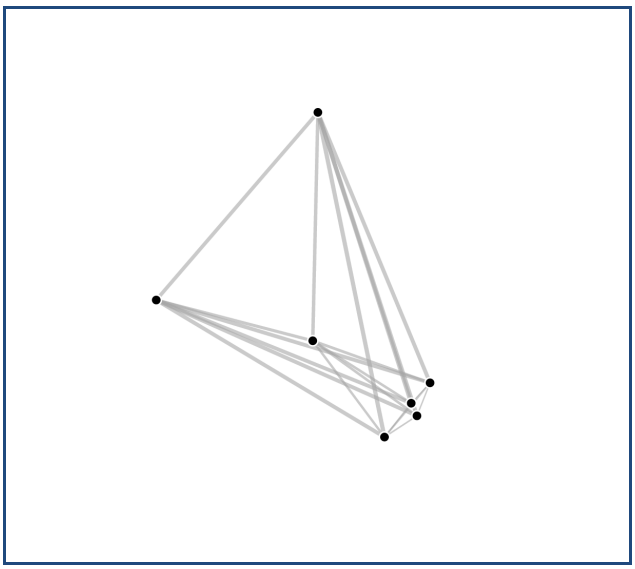

Figure 24: A connected graph based on the Arial font representing similarity scores of Latin, Greek, Cyrillic, Hebrew, Arabic, Thai and Devanagari. The length of the edge correspondes to the similarity score.

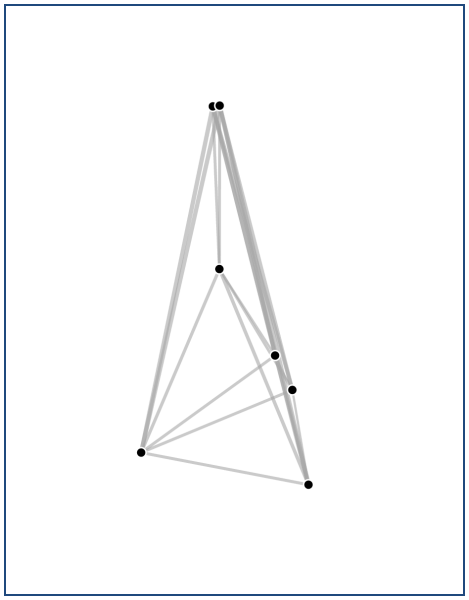

Figure 25: A connected graphs based on the Courier New font. The top cluster, which contains Thai and Devanagari, does not exist with the Arial font analysis and the distances between Latin, Greek, Cyrillic and Hebrew are greater.

4.8 Heatmap

Since our focus is on the comparison between the scripts, and evaluating edge length seemed less intuitive, as well as the sever problem of occlusion that would have interfered with an easy interpretation of the relationships. Therefore we decided to use heatmaps instead. The heatmap is displayed in Figure 26. In order to display our data in a heatmap we generated the similarity score for every combination of the scripts – 55 values - all scores are presented in Table 3. We created the heatmap based on an example by Damian Kap, presented in a blog post[9]. The intensity of the color corresponds to stronger similarity between the scripts that intersect on a specific rectangle – white representing the greatest disparity and the strongest green on the diagonal shows the identity of a script with itself. These rectangles are linked to the Comparison section described in

[9] http://blog.nextgenetics.net/?e=44

Section 4.9. At the bottom of heatmap a gradient scale is displayed. Selecting a rectangle also marks the similarity score on this scale.

Since the number of scripts is fairly low and there was an immediate correlation with the evolution tree, the scripts were manually sorted based on their similarity score. It should be interesting to run a hierarchical clustering algorithm on the scripts themselves to see how well such an algorithm corresponds to the evolution tree.

Script 1	Script 2	Similarity Score	Script 1	Script 2	Similarity Score
Greek	Latin	0.96	Hebrew	Arabic	0.78
	Cyrillic	0.98		Thai	0.43
	Hebrew	0.88		Tamil	0.38
	Arabic	0.73		Malayalam	0.33
	Thai	0.39		Telugu	0.063
	Tamil	0.34		Gujarati	0.43
	Malayalam	0.27		Devanagari	0.45
	Telugu	0.00	Arabic	Thai	0.64
	Gujarati	0.39		Tamil	0.59
	Devanagari	0.50		Malayalam	0.55
Latin	Cyrillic	0.95		Telugu	0.27
	Hebrew	0.92		Gujarati	0.65
	Arabic	0.75		Devanagari	0.58
	Thai	0.40	Thai	Tamil	0.95
	Tamil	0.36		Malayalam	0.84
	Malayalam	0.30		Telugu	0.59
	Telugu	0.023		Gujarati	0.99
	Gujarati	0.41		Devanagari	0.58
	Devanagari	0.49	Tamil	Malayalam	0.84

Cyrillic	Hebrew	0.87		Telugu	0.61
	Arabic	0.74		Gujarati	0.94
	Thai	0.40		Devanagari	0.58
	Tamil	0.36	Malayalam	Telugu	0.73
	Malayalam	0.29		Gujarati	0.85
	Telugu	0.014		Devanagari	0.43
	Gujarati	0.41	Telugu	Gujarati	0.59
	Devanagari	0.52		Devanagari	0.20
			Gujarati	Devanagari	0.58

Table 3: The 55 similarity scores generated between the 11 scripts. Based on these scores the color intensity in the heatmap is defined.

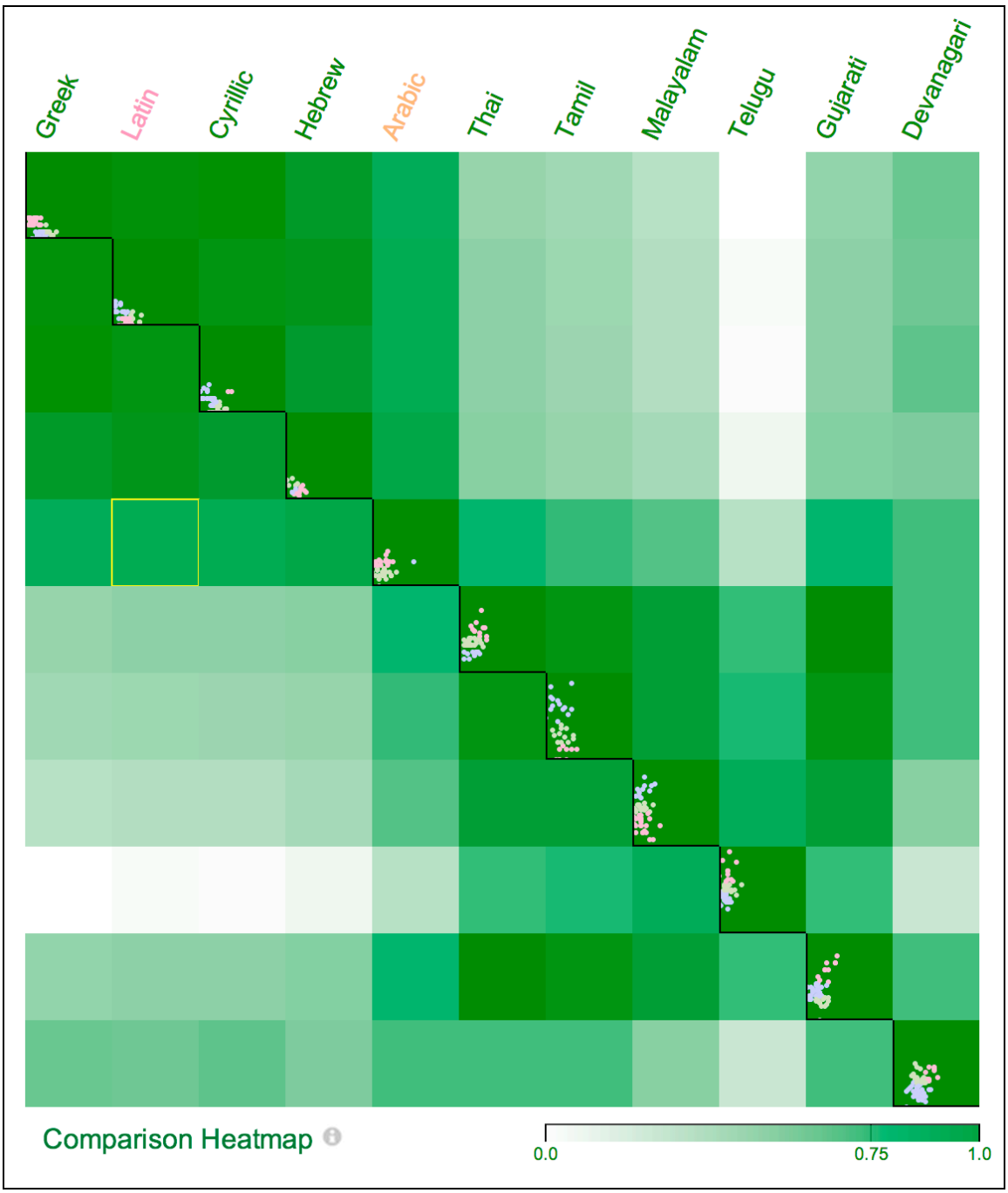

Figure 26: The heatmap showing the similarity score between the 11 scripts. This image shows the heatmap without the overlaying scatterp plots along the diagonal that exist in the visualization.

4.9 Comparison Section

The comparison section is meant to provide extra data when comparing two scripts. It is linked to the heatmap so selecting one of the rectangles in the heatmap, brings up the information on the corresponding scripts (which intersect in that rectangle). The idea is to provide insight into the two scripts and what contributes to their similarity score. Therefore a significant part of the exploration is done in this section. The name of the script at the top is a link to its Wikipedia page to allow users to further investigate a script. Below that we display a scatter plot in which clusters and representative characters are shown. A separate discussion on the scatter plots can be found in Section 4.10. At the bottom of this section a map of the world is presented with the countries in which those scripts are used are highlighted. The information for this map was manually compiled based on the Wikipedia article on Writing systems[10], which was created and updated based on articles and books on the topic (Coulmas, 1999), (Daniels &

[10] http://en.wikipedia.org/wiki/Writing_system

Bright, 1996), (Sampson, 1985). Though the different scripts of India are mostly used in different subregions of the country, we decided to mark the entire country for all script since their proximity is great when viewing on a global scale, and the development overhead for impleneting subregions seemed unnecessary.

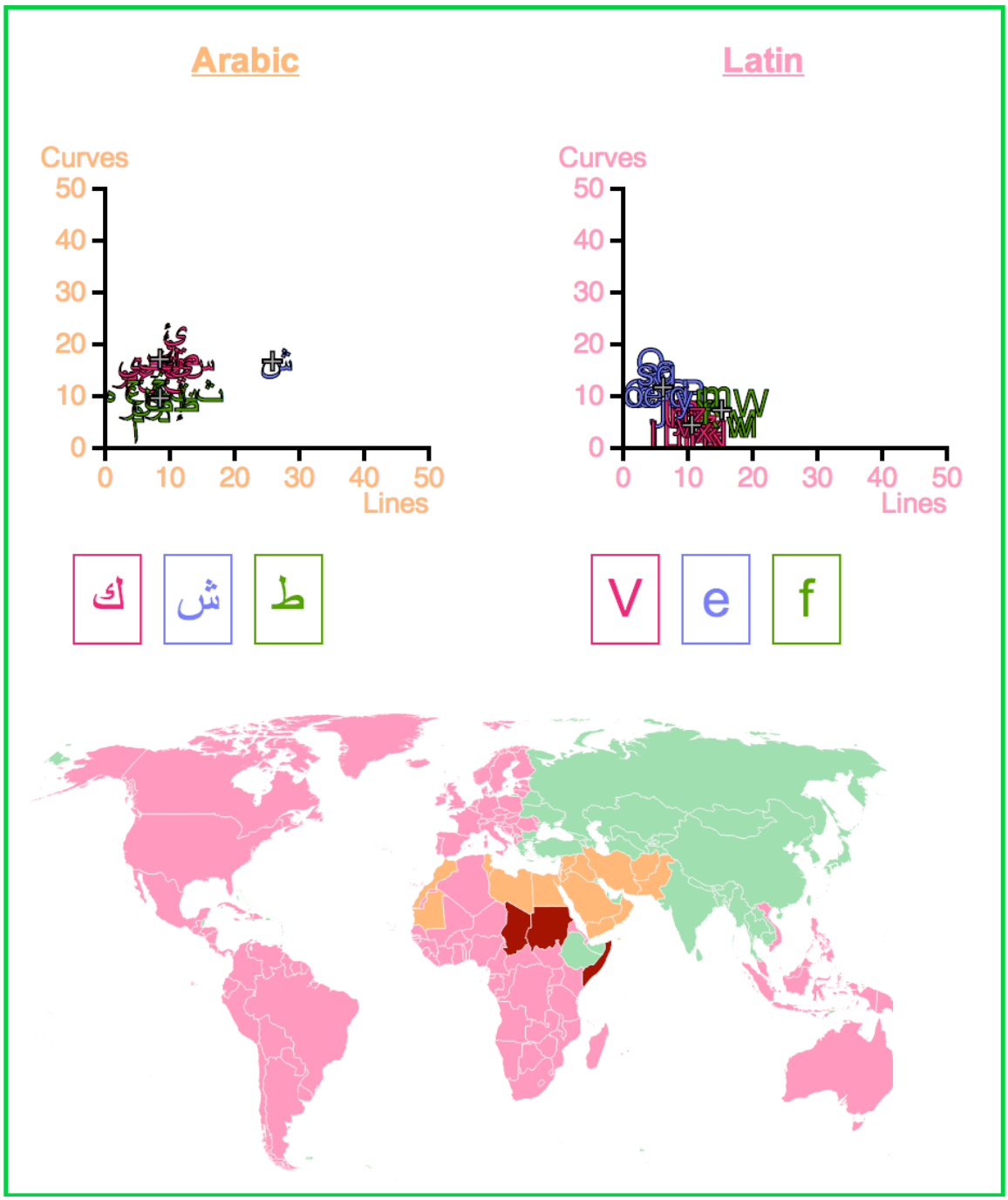

Figure 27: The comparison section of the visualization showing the comparison of Latin and Arabic. The scripts names at the top are links to the Wikipedia articles. The scatter plots show the distribution and clusters of the letters as well as the representative characters. The scale shows the similarity score between Latin and Arabic – 0.75. The world map shows the countries which use a Latin derived

alphabet (pink), the countries that use the Arabic script (orange) and countries that use both (brown).

4.10 Letters Scatter Plots and Clustering

Visualizing the letters in a scatter plot provides valuable insight into the algorithm and the similarity or disparity between scripts. The axes are the number of lines and the number of curves per character. This shows not only the difference in ratio between lines and curves, but also the total number of segments affects the similarity score. If we examine the scatter plots in Figure 28 we can easily see that Latin and Arabic has much less lines in general and less curves specifically. While a character in Latin will have less than 30 segments, characters in Telugu go up to 50 segments and more, most of which are curves. These differences translate into the similarity score, which is 0.75 between Latin and Arabic, vs. 0.023 between Telugu and Latin and 0.27 between Telugu and Arabic.

In order to derive representative characters for each script we have run a clustering algorithm on these data points, using the number lines and curves as our coordinates. Since the classic kmeans clustering algorithm has a high risk of falling into local minima, we have decided to use a bisecting kmeans algorithm as described by Peter Harrington (Harrington , 2012). Once we found the clusters, using the Euclidean distance function described in Section 4.5, we used the

clusters centers in order to find the character closest to it. Each cluster is in a different color and the centers are marked with a gray +.

Since these clusters represent different “areas” of the letters, we perceive them as having varying characteristics; therefore we chose them to represent the script. We found that having three representative characters provided the most useful insight into the elements of the script. Having only two representative clusters did not provide enough sample and presenting four characters or more was too much and introduces redundancies.

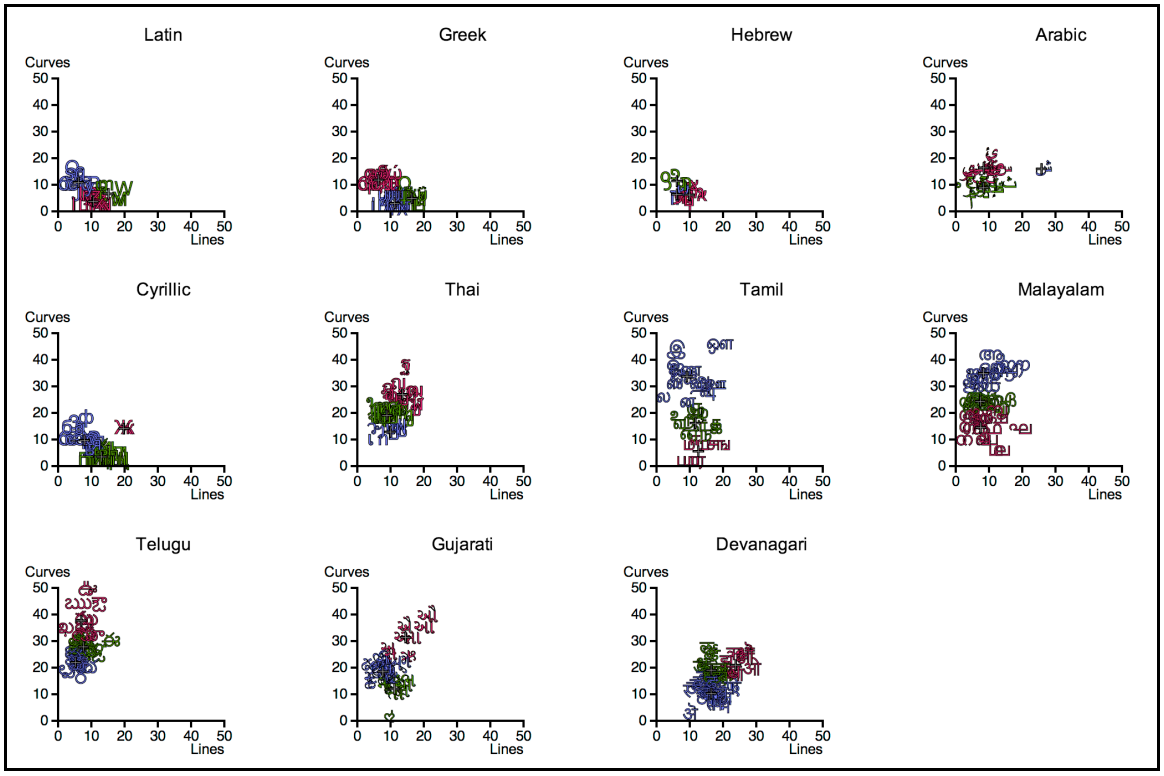

Figure 28: Scatter plots of all the scripts. This small multiple shows the difference in distribution which translates into a higher similarity score between the scripts.

Chapter 5 – Summary and Conclusions

This chapter describes conclusions from the data exploration and visualization, discussion about lessons that were learned working on this thesis and future work that can be done on the topic.

5.1 Discussion

The first challenge we had to deal with was the size and fuzziness of the data. There are a great number of scripts in use today and the historical information on each is staggering and poses an interesting problem of data collection. Since this data is not numeric but is more of a collection of narratives, gathering this data is not straightforward. We had to combine web scrapping, manipulating downloaded and gathered data through scripts and manually and other methods in order to obtain the data used in this project. In addition we had to keep our focus and decide what data to explore and represent and which to put aside. This includes the decision to focus on scripts instead of writing system, the choice of scripts that derive from a single origin and focusing on one font. While all of these options pose interesting opportunities, we decided to avoid spreading too much. Another related challenge was the lack of prior work. This meant we had to gather and explore everything from scratch and constrained our

scope even further. All of these challenges encouraged us to limit the scope and focus on a subset of the data and a narrower analysis.

Once we defined the data, we had to consider how best to visualize it. Instead of having multiple views into the data separated into multiple tabs or windows, we decided to condense all the information into a single linked view. This allows the user to explore, compare and investigate within the same context. While heatmaps are less familiar to users, they allowed for a simple display of the information and moreover a linking of a comparison (between two scripts) becomes intuitive.

5.2 Limitations

The biggest limitation we see is the simplicity of the algorithm, measuring only average number of lines and number of curves. Observing the similarity scores in Table 3 there are several that reach 0.99 or 0.98. We find this score to be too high and reflect the limitation of the algorithm. Gujarati and Thai for example, are not similar enough to the eye to be evaluated as a 0.99 similarity. This is caused by the closeness of the average measurements – 8.25 / 8.4 lines and 17.41 / 17.52 curves to Gujarati and Thai respectively.

As a start, the length of the different segments and convexity of the curve can be taken into account. It is not only the existence of a line or a curve that influence the visual themes of scripts, but also their length and extremity. Another

parameter we have considered is the density of the glyphs, that is counting the number of dark vs. light pixels. For example Tamil tends to have very narrow glyphs. The variance inside a script could also form an interesting input. If we define a distance function between glyphs, such as the Euclidean function used, we can measure the average distance between characters and take that into account in our similarity score. Lastly, a simple measurement of number of characters may also yield improvement in similarity score ranking.

5.3 Conclusions

The strongest most obvious conclusion was the strong correlation between the similarity clusters and the evolution tree as well as geographic information. The heatmap clearly shows two clusters, one containing the scripts closely related to Europa and the other containing the Indian and other far east scripts from the Brāhmī family (which we will refer to as the “west” and “east” cluster for ease of use). While this correlation is somewhat expected it was still exciting to see it present itself so clearly in our results and it served as some validation of our algorithm. Another interesting data point was Arabic, which serves almost as a bridge between the two clusters, with almost all of its intersections comparing above 0.5. This can be explained both geographically – since it is common in the countries between Europe and India – and by looking at historical data. In ancient times Arab traders were the main bridge between the

two worlds, transporting goods (silk and spice) from east to west (XXX reference).

Looking at the small multiple of the clusters we notice that the “west” cluster has a tendency to have less curves compared to straight lines and less segments in general. The “east” cluster is much more curved and tends to have more segments. A couple of outliers that may require more exploration are Devanagari and Telugu. Devanagari, though it is geographically and historically close to the other Brāhmī scripts, scored no more than 0.58 and not much lower in scores with the “west” cluster. The high number of segments on one hand and the high number of straight lines on the other is the cause. Telugu on the other hand, while it is relatively close in comparison with the other “east” scripts, shows the greatest difference with the “west” cluster. Again looking at the small multiple we find that Telugus characters are furthest from the origin, which translates into very complex characters. Every character in Telugu as at minimum 20 segments, and most of the segments are curves. The combination of these two features is what makes Telugu so “far” from the “west” cluster.

5.4 Future Work

We consider this project the very beginning of a journey into exploring scripts from a visual perspective. There is much work that can be done, both as a continuation of this project directly and new projects stemming from this one.

First, this project can be extended to contain more scripts. It will be particularly interesting to see a comparison with scripts that do not stem from the Egyptian Hieroglyphs origin such as the Chinese character set. In addition a comparison to scores gathered using different fonts in general and Serif scripts in particular (unlike the Arial San-Serif font used) may lead to very interesting results. Some other features that can be added include the ability to compare more than two scripts and more information upon comparison. We would have like to add the ability to brush entire sections of the heatmap to compare multiple scripts, however this will also require rethinking of the comparison area. We would have also liked to add more data in the comparison such as the estimated time in history the script was created. Last but not least, instead of using representative characters, a machine-learning algorithm can be written to learn the characteristics of a certain script and produce a generic character based on it. *XXX Add reference to generic numeral creation research*. This type of algorithm may extract the most crucial visual elements of a script and highlight its visual themes.

Next projects in this vein can include a good visualization of the development of different scripts from shared origins. This holds both a linguistics interest and a fascinating visualization challenge to express visual development in multiple concurrent paths. It will also make an interesting project to attempt the problem of writing systems and not scripts. This opens many questions, both from data collection and analysis as well as visualization and focus. Such a project will also be able to build upon fascinating existing work done in the field on comparing different European writing systems. Further from this research and harder to implement, lies the idea of comparing ancient origins. The origins of modern scripts tend to be close to picture writing. We feel a compelling visual opportunity can be found in comparing Egyptian Hieroglyphs, Oracle Bone Scripts and the Mayan Hieroglyphs for example. Each of these scripts has distinct visual themes and a captivating historical context, which can form the ground for interesting explorations and visualizations.

5.5 A Personal Note

We have set out to this journey with little knowledge but tremendous curiosity towards scripts and writing systems. We have found intriguing scripts, surprising connections and interesting visualizations. While a lot was learned and discovered working on this project, we feel the voyage into exploring scripts and the roots of their visual themes has just begun. This topic, which touches upon linguistics and data visualizations, has an enormous potential in our mind. The

breadth and depth of the knowledge, the questions that are waiting to be asked and the visual world that is waiting to be explored, poses exciting opportunities for visualization.

References

Abdul-Rahman, A., Lein, J., Coles, K., Maguire, E., Meyer, M., Wynne, M., et al. (2013). Rule-based Visual Mappings – with a Case Study on Poetry Visualization. *Eurographics Conference on Visualization (EuroVis).* Leipzig: Computer Graphics Forum.

Abou Rjeily, R. (2011). *Cultural Connectives.* New York: Mark Batty Publisher.

Bringhurst, R. (2004). *The Elements of Typographic Style.* Vancouver, Canada: Hartley and Marks Publishers.

Collins, C., Carpendale, S., & Penn, G. (2009). DocuBurst: Visualizing Document Content Using Language Structure. *Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis).* Berlin: Computer Graphics Forum.

Coulmas, F. (1999). *The Blackwell Encyclopedia of Writing Systems.* Oxford: Wiley-Blackwell.

Daniels, T. P., & Bright, W. (Eds.). (1996). *The World's Writing Systems.* Oxford University Press, USA.

Fradkin, R. (2000, February 10). *Evolution of Alphabets*. (C. Seljos, Producer) Retrieved March 6, 2014, from The University of Maryland: http://terpconnect.umd.edu/~rfradkin/alphapage.html

Gell-Mann, M., & Ruhlen, M. (2011). The origin and evolution of word order. *Proceedings of the National Academy of Sciences of the United States of America* .

Ghoniem , M., Fekete , J.-D., & Castagliola, P. (2005). On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization* , 22.

Harrington , P. (2012). Grouping unlabeled items using k-means clustering. In P. Harrington, *Machine Learning in Action.* Shelter Island, NY, USA: Manning Publications Co.

Heller, M. (2012, February 18). *Wordcollider.* Retrieved February 24, 2014, from Vimeo: http://vimeo.com/37015401

Rufiange, S., McGuffin, M. J., & Fuhrman, C. P. (2012). TreeMatrix: A Hybrid Visualization of Compound Graphs. *Computer Graphics Forum , 31*, 89-101.

Sampson, G. (1985). *Writing Systems.* Stanford, USA: Stanford University Press.

Seo, J., & Shneiderman , B. (2002). Interactively Exploring Hierarchical Clustering Results. *Computer , 35* (7), 80-86 .

Szczepanski, A., & Meaney, E. (n.d.). *Info*. Retrieved February 24, 2012, from Null Sets: http://evanmeaney.com/ns/index.html

Tyshchenko, K. (2000). *The Metatheory of linguistics.* Ukrein: Osnovy.

Wilkinson, L., & Friendly , M. (2009). The History of the Cluster Heat Map. *The American Staticians , 63* (2), 179–184.

Yardeni, A. (1993). *HarpatkaOT.* Jerusalem, Israel: Karta.