Reviews in Digital Humanities • Vol. 2, No. 12

Review: Dante Visualised

Crystal Hall¹

¹**Bowdoin College**

Published on: Dec 13, 2021

DOI: 10.21428/3e88f64f.7b47d8a1

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Project

Dante Visualised

Project Director

[Ginestra Ferraro](#), King's College London

Project URL

<https://github.com/ginestra/dante-visualised>

Project Reviewer

[Crystal Hall](#), Bowdoin College

Project Overview**Ginestra Ferraro**

Dante Visualised explores how to develop a reusable tool to generate data visualizations based on automatic text analysis. Existing tools such as [Voyant Tools](#), [HuViz](#), and [iteal](#) either collate a wide range of existing tools or target very specific cases. The aim of Dante Visualised is to provide a lightweight and solid application core that is extensible in the hands of the user but doesn't lose its simplicity in use at setup. Flexible in nature, the tool accepts different text inputs and is optimized to produce rich visualizations with minimal setup. The visual outputs produced by the application aim to offer a different perspective on a text under study; highlight patterns and/or outliers (Meirelles 2013), and drive research in formulating new hypotheses and supporting or disprove existing theses.

The current version is designed around Dante Alighieri's *Divine Comedy*, but it serves as a blueprint for further components. The unique way Dante wrote his masterpiece makes the text an interesting dataset to explore computationally. Structural (spatial and temporal) textual components lend themselves to graphical representation, which offers insights into its linguistic content. The Italian version of the *Commedia* (Petrocchi 1966-67) demonstrates the tool's utility for text structural analysis and work on the rhyme scheme, while the English translation (Mandelbaum 1980-84) is used for sentiment analysis. The visual outputs allow users to interact with both content and metadata.

The application performs computational text analysis to produce data visualizations representing the following structural, stylistic, and semantic features of the text:

schematic representation of the poem's structure and rhythm (fig. 1), distribution of keywords (fig. 2), and visual representation of the sentiment analysis (fig. 3).

Figure 1 An example of the schematic representation of the poem's structure: rhythm imposed by tercets and rhyme prediction.

Figure 2 Words like *Cristo* (Christ) and *stelle* (stars) are distributed unevenly across the three cantiche; the word “Christ” never appears in the *Inferno*, while it’s widely used in the *Paradiso*. Each square corresponds to a line.

Figure 3 Sentiment analysis visualization of the three cantiche. Red is negative, blue is positive, and the opacity indicates how close to the polarity (-1, 1) the sentiment is. Each square corresponds to a line.

The application has been developed modularly (Martin and Martin 2006), following the “separation of concerns” design principle (Dijkstra 1982) to allow for flexibility and scalability. It is implemented in Python, a flexible programming language that supports object-oriented programming and functional paradigms. The visualizations are

produced with the support of the d3.js data visualization library. The application exploits HTML5 and SVG specifications to allow for greater interaction and portability.

Natural language processing and machine learning techniques process and transform the data. We use the Naive Bayes Classifier (Perkins 2010) technique due to its performance and simple implementation. Further, we developed a training dataset by collecting random subsets of text from additional work that is close in language and time, including Ludovico Ariosto's *Orlando furioso*, Dante's *Convivio*, and Giovanni Boccaccio's *Decamerone*.

The application's modular structure (fig. 4) makes it amenable to further development (e.g. algorithm refinements, visualization workflows, and stylometric analysis).

Figure 4 The data model of the application, illustrating the separation of concerns and the potential for extensibility.

This work is the result of a final MSc Computer Science project developed by Ginestra Ferraro, built using 10% personal development time offered by the King's College Digital Lab and the developer's evenings and weekends for two months during the summer of 2018. The project is irregularly updated and is still in its embryonic stage as a proof of concept with a vision to create a reusable tool to generate semi-automated data visualizations based on text analysis.

Works Cited

Dijkstra, E. W. (1982). "On the Role of Scientific Thought." *Selected Writings on Computing: A Personal Perspective*. Springer-Verlag, pp. 60–66.

Mandelbaum, A. (1980-84). "Divine Comedy of Dante Alighieri." University of California Press.

Martin R. C., Martin M. (2006). *Agile Principles, Patterns, And Practices in C#*. Prentice Hall.

Meirelles I. (2013). *Design for Information: An Introduction to the Histories, Theories, and Best Practices Behind Effective Information Visualizations*. Rockport Publishers.

Perkins J., (2010) "Text Classifier for Sentiment analysis - Naive Bayes Classifier," May 10. <https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/>

Petrocchi, G. (1966-67). "Dante Alighieri, La Commedia secondo l'antica vulgata, a cura di Giorgio Petrocchi." Mondadori.

Project Review

Crystal Hall

Dante Visualised is an interactive, visual platform for pattern exploration tools that examine structured texts. The site takes its name from the Italian author Dante Alighieri (1265-1321), whose *Divine Comedy*, in Italian and English versions, provides the case study for the tool development and proof-of-concept. The project exists as a downloadable Git repository of the code for rendering the visualizations and the interactive companion website built from it. Dante's poem has been subject to a significant number of digital interventions, yet Dante Visualised breaks new ground through its emphasis on visual analysis of the text.

The selection of Dante's text provides a valuable opportunity to both learn more about a complex poem and to push computation further into the service of asking humanistic questions. While remaining philological, the added distance between the viewer and the poem creates an important analytical space. The text itself is computational, following a rigid structure of interlocking verses (*terza rima*) and nested episodic segments (3 canticas that contain a total of 100 cantos). The tools provide an opportunity to interpret graphic information derived from the text: color and line length. By abstracting from alphabetical characters to visual properties, the user is drawn more quickly to patterns (or the absence thereof) in the structure of the poem. The visualizations address questions of the consistency of rhymes, lexical choices, line lengths, and positive or negative sentiment.

The tools have been designed for eventual application to texts beyond the *Comedy*. A planned feature will allow users to upload a local text file, select structural

parameters, and submit it for server-side processing via Python to generate a JSON object that would drive the JavaScript visualization using the D3.js data visualization library. Currently, sentiment analysis of the text is limited to English, using Natural Language Processing and the Naïve Bayes classifier that are part of the NLTK Python package. Further documentation would be necessary to understand how the training texts were tagged and any iterative processes used to refine the resulting model. The training set itself will be a valuable resource for digital humanities projects involving translations of early Italian texts. In the proposed expansion of the project, the advantage over Voyant Tools is that the user will be able to provide or create a text file with added metadata for analysis, relying less on lemmatization and word frequency for insight.

The use of automated, text-informed RGB value setting for the rhyming syllables provides a novel lens for investigating the sounds of the poem. The ASCII codes for each of the final three characters of a line are used to assign the color values, which immediately draw attention to understudied rhyming strategies in the poem, while dulling the distinctions between those areas that have received more critical attention (i.e. Sicilian rhymes that rely on accented characters, all of which are nearly sequential in the ASCII coding schema). This technique could have valuable applications beyond the *Comedy*.

Overall, Dante Visualised offers the kind of global view of a lengthy and complex text that would be an immediately helpful resource for students studying the poem. In terms of wider use, the future extensibility will allow for ground-breaking comparative research in Dante studies of translations and variants via the rhyme and sentiment output. Possibilities seem equally promising for use in later periods and literature in other languages.