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.

The computational aspect of the project is implemented in Python, a flexible programming language that supports object-oriented programming and functional paradigms.

The visualisations are produced with the support of D3.js library, a JavaScript library for manipulating documents based on data” (Bostock 2011). The application exploits HTML5 and SVG specifications to allow for greater interaction and portability.

Natural language processing (NLP) and machine learning techniques have been applied to process and transform the data. The Naïve Bayes Classifier (Perrin 2010) technique has been chosen due to its performance and simple implementation. A training dataset has been manually created collecting random subsets of text from other authors close in language and time, and further work from Dante himself:

- offer a different perspective on the text under study;
- highlight patterns and/or outliers (Meirelles 2013);
- drive research in formulating new hypotheses;
- provide support to, or disprove, existing theses.

The current version accounts for modules (i.e. software components) designed around one selected text case, namely Dante Alighieri’s Divine Comedy, but serves as a blueprint for further modules to be plugged in.

The main success lies in its modular development (fig. 4), making it amenable to further development (algorithm refinements, visualization workflows, stylometric analysis). More languages and different text structures will be integrated and a wider range of output visualizations offered, while making use of the same core functionalities for ingesting and processing data.

The Italian version of the Commedia (Petrarca 1966–67) is used to perform text structural analysis and work on the rhyme scheme, while the English translation (Mandelbaum 1980–84) is used for sentiment analysis. The unique way in which Dante wrote his masterpiece, makes the text an interesting dataset to be explored computationally. Structural (spatial and temporal) textual components lend themselves to be represented graphically, and offer insights into its linguistic content. The visual outputs allow users to interact with both the content and the metadata.

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

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

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

The project’s software is offered, while making use of the same core functionalities for ingesting and processing data.

The application has been developed:
- to allow for flexibility and scalability;
- for natural language processing (NLP) and machine learning techniques.

The computational aspect of the project is implemented in Python, a flexible programming language that supports object-oriented programming and functional paradigms.

The visualisations are produced with the support of D3.js library, a JavaScript library for manipulating documents based on data" (Bostock 2011). The application exploits HTML5 and SVG specifications to allow for greater interaction and portability.

Natural language processing (NLP) and machine learning techniques have been applied to process and transform the data. The Naïve Bayes Classifier (Perrin 2010) technique has been chosen due to its performance and simple implementation. 

A training dataset has been manually created collecting random subsets of text from other authors close in language and time, and further work from Dante himself:

- offer a different perspective on the text under study;
- highlight patterns and/or outliers (Meirelles 2013);
- drive research in formulating new hypotheses;
- provide support to, or disprove, existing theses.

The current version accounts for modules (i.e. software components) designed around one selected text case, namely Dante Alighieri’s Divine Comedy, but serves as a blueprint for further modules to be plugged in.

The main success lies in its modular development (fig. 4), making it amenable to further development (algorithm refinements, visualization workflows, stylometric analysis). More languages and different text structures will be integrated and a wider range of output visualizations offered, while making use of the same core functionalities for ingesting and processing data.

The Italian version of the Commedia (Petrarca 1966–67) is used to perform text structural analysis and work on the rhyme scheme, while the English translation (Mandelbaum 1980–84) is used for sentiment analysis. The unique way in which Dante wrote his masterpiece, makes the text an interesting dataset to be explored computationally. Structural (spatial and temporal) textual components lend themselves to be represented graphically, and offer insights into its linguistic content. The visual outputs allow users to interact with both the content and the metadata.

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

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

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

The project’s software is offered, while making use of the same core functionalities for ingesting and processing data.

The application has been developed:
- to allow for flexibility and scalability;
- for natural language processing (NLP) and machine learning techniques.