Master’s Degree programme

in Language Sciences

Final Thesis

Computational analysis of the prosodic characteristics of the poets’ voice

Supervisor

Dr. Gianluca E. Lebani

Assistant supervisor

Ch. Prof. Alessandro Mistrorigo

Graduand

Álvaro Macías López

Matriculation Number 893604

Academic Year

2023/2024

Abstract

This thesis presents the development of a novel computational tool, PAUSE (**P**honodia´s **A**nnotation and vis**U**lization of poet**S**' voic**E**), designed to explore, and analyse the intricate prosody of poetry. Using different state of the art techniques, the tool aims to unveil the rhythmic and melodic patterns inherent in poetic expression, allowing for the study of their role in conveying the emotion and aesthetics of poetry.

The project begins with a comprehensive review of existing literature on prosody, computational linguistics, and their intersection in the realm of literary analysis. Emphasizing the significance of the voice in understanding the nuances of poetic form, this research sets out to bridge the gap between traditional literary analysis and cutting-edge computational methodologies.

Pause utilizes tools that have been used extensively in previous research in the field to dissect the acoustic features of audio recordings, capturing the intensity, pitch, and speed that contribute to the unique prosodic fingerprint of each poem. They include algorithmic solutions as well as machine learning models that enable us to segment and analyse spoken language. These tools have proven their reliability and accuracy on the aforementioned research.

The implementation of Pause is accompanied by a user-friendly interface, facilitating accessibility for scholars and enthusiasts interested in exploring the prosodic intricacies of poetry. The tool allows users to input poetic texts alongside recordings of those poems, visualize prosodic patterns, and generate insightful analyses. The results are different annotations of the provided text that can be used for a critical study of poetry. These annotations are accompanied by the relevant data to explain and contextualize them.

Through a series of use cases, this thesis also provides the user with an insightful guide on the use of the tool, as well as providing examples of its potential use in education and outreach.

In conclusion, Pause represents a significant contribution to the field of computational linguistics applied to literature. This tool facilitates the work of researchers approaching a critical study of the prosodic components of poetry, allowing for a close reading of the material, and yielding measurable and replicable insights.

Acknowledgements

I could not thank enough my supervisors, Dr. Gianluca E. Lebani and Ch. Prof. Alessandro Mistrorigo, for their support and advice during the development of this project, as well as their flexibility that allowed me to perform my research from abroad.

Finally, I would like to thank all the colleagues, friends and professors I have met during my years in college and have been an invaluable support all the way.

Table of contents

Figures

Tables

1. Introduction and reach

This project is born with the goal of providing a tool to support the work of researchers in the fields of literary criticism and phonetics that wish to approach the study of poetry – in particular, of the phonetics and prosodic elements of poetry reading – from a quantifiable point of view. The inclusion of language tools in linguistic fields has proven as a reliable method for rapidly and precisely dealing with the manual and mechanical work, which traditionally required either large amounts of people or a huge time investment. The manual annotation of texts also comes with the downside of cohesion between annotators, as even a small variation in the application of the annotation guidelines can lead to inconsistencies in the results. What we aim for, then, is creating a tool that automatizes the process of annotation, that allows for the distant-reading, precise and granular analysis that a computer is capable of, alongside the complex, literary, specific, and sensitive analysis that the human reader is capable of. To the best of our knowledge, a tool like this one does not already exist.

This work, then, is primarily aimed at researchers. But because of the nature of these researchers' production, the project is developed with outreach in mind. This way, this tool should serve not only in an academic environment, but as a way to disseminate the researcher's work and way of looking at and understanding poetry. The annotations of this tool could be used for educational porpoises at all levels. It will allow for analysis of custom complexity, giving the possibility of creating different sets of material from the same poems, adjusting them to the needs of different classes. It is ultimately aimed, then, at anyone with an interest in poetry criticism.

1.1. Definitions

In this section we will provide definitions for the different terms that come from the technical languages of fields other than linguistics. Due to the multidisciplinary nature of the projects, we consider it necessary to provide readers with these definitions in order to ensure that there is a common notion of their meaning, and to disambiguate if needed.

An **enjambment** (Baldick, 2008) is the running over of a verse structure to the next one without a punctuated pause. Although this classical definition refers often just to the grammatical structure, it is only natural to extent the definition to apply to prosodic structure, too. An enjambment, then, is the running over of the grammatical or prosodic structures of a verse to the next one, without pausing at the end of the first verse of the two. In this project we will generalize and extent this definition. Will consider also “running over” a punctuation mark as an enjambment. A punctuation mark in the usual prosody of Spanish comes with a pause in speech. Hence, we consider that not doing a pause where one would expect it due to a punctuation mark is a kind of intra-verse enjambment.

In modern prosody, a **caesura** (Britannica, 2016) is a pause within a verse that breaks the regularity of the metrical pattern. It may fall at any point of the verse. It can be thought of as an unexpected pause or break of the verse, as it interrupts its metrical pattern. In poetic analysis and criticism, it is often marked with a double vertical line ||. Although we can consider it as an irregularity because it goes again the rhythmic pattern, it can be used to introduce the otherwise usual cadence of natural speech. We will consider different types of caesuras depending on where they fall within the verse or within the grammatical structure.

Now we will define some important terms related to computer science. Although some of them may seem trivial to those with experience in computer science, the goal is to give a formal definition of them to ensure that all readers have the necessary common ground.

The main problem that we work on, from a computational point of view, is called **speech recognition**. Usually referred to as speech-to-text, the term includes the field and set of technologies that process human speech to work on it in written form (Cole et al., 1997). It can refer to the process of creating a text from the audio signal, or, as in our case, to the process of establishing a relation between an audio signal and a text to analyse it (Moreno et al., 1998). In order to establish this relation, a certain technique known as forced alignment must be applied. **Forced alignment** is the procedure to create a bidirectional mapping between a speech recording and the corresponding text (Li et al., 2022). It is a complex technique usually involving machine learning models to create the software capable of applying it.

We have mentioned machine learning, and we will discuss its possible applications in this project in the following sections. It is appropriate then to give a proper definition of the term. **Machine learning** is a field of artificial intelligence that develops algorithms capable of drawing inferences and patterns from a set of data – called the training set – and use those patterns to make predictions or classifications on unseen data (Zhou, 2021). Informally, we could define it as a set of algorithms that are able to learn from the data they are shown, and then apply that knowledge to predict something from data that they have never seen before. The power of these techniques is that by applying these methods we can create software capable of solving complex problems without specifying explicit instructions on how to solve them. The output of a machine learning algorithm is called a **model**. Models are the mathematical expression of the inferences made by the algorithm and that are applied to a data set in order to make predictions (Burkov, 2019). The main hardship when using machine learning is training the model adequately, which usually is a task to be performed for each particular problem. In the case of computational linguistics this means that a model serves only for one specific language or task. Training a model – creating quality and exhaustive data sets from which the need inferences can be drawn – is a complex and expensive task. There is, however, a technique called **fine-tuning** that allows to reduce the cost and the time need to train a model. This approach consists of using a pre-trained model, usually for a general task, and train it at a lower scale for a more specific task (Quinn, 2020). In our case, this could allow for efficient use of large language models, tweaking them for our use cases.

Due to the difficulties mentioned in the previous paragraph, we will not be using machine learning in this project. Instead, we will be developing algorithms and heuristics to process the data. **Algorithms** (Dasgupta, Papadimitriou and Vazirani, 2006) are sequences of instructions that are precise, unambiguous, mechanical, efficient and correct to solve a problem or perform a task in a finite number of steps. They can be thought as a cooking recipe for computers.

Heuristics (Apter, 1970) are techniques used to find approximate solutions to problems when the exact solution would take a prohibitively long time to be found. In a large number of use cases, the best possible solution for a problem is not required, or it is not feasible to obtain due to the time it would require doing so. In these cases, approximations provided by heuristics are a great compromise between accuracy and execution time.

Audio signals contain huge amounts of information. So much, in fact, that we cannot process all of it, and even if we could, most of it is not relevant. For these reasons, we first have to remove some amount of data for our algorithms to work. The process of reducing the amount of data we have to work with is called **data compression** (Mahdi, Mohammed and Mohamed, 2012). It is a technique to reduce the size of data in order to save space or transmission time. In particular, we will use **lossy compression** (Wolfram, 2002). This type of compression assumes that some data is going to be lost to reduce its size. It performs a trade-off between keeping all the information and reducing the size, so nonessential details are dropped.

To build out tool, we will rely on Application Programming Interfaces (**APIs)** (Reddy, 2011). APIs are a way to connect computers or computer programs. They are used by developers to add someone else’s functionalities to their programs without having to know the internal details of how they work. This way, we can reuse existing code in our tool.

1.2. State of the art

The scope of this project covers multiple fields, mainly in the frames of linguistics, computer sciences and literary studies. It could be defined as a digital tool for humanities due to its goal of aiding the researcher with the study of written texts. As well as a computational linguistics project, for its methodology of automating the extraction and analysis of linguistic data from written and spoken sources. Because these analyses are carried out with the intention of annotating text, the project could fall under the definition of two broad and well-known problems in computational linguistics: speech-to-text (also known as STT) and automatic annotation. Furthermore, with its focus on poetics it arguably falls into the fields of cognitive poetics and computational stylistics, an intersection of poetry criticism and computational linguistics. Additionally, it requires a background in phonetics from which to draw the necessary theory to analyse and contextualize the audio input and its physical components. Finally, some insight in literary studies is needed to properly assess the need for this tool in the field. This means that our bibliography needs to cover the state of the art of those fields to properly contextualise this dissertation.

When it comes to computational linguistics, specifically the field of Natural Language Processing (NLP) is currently dominated by Artificial Intelligence (AI), or, more in particular, Machine Learning (ML). A particular subset of ML technologies, known as Deep Learning, has brought great advances to the field of NLP. It has allowed for a broadening of the capabilities and applications of NLP. Some of these applications are an enhancement of some tasks closely related to linguistic fields. For example, text summarization. This is a task that requires a degree of semantical analysis, which at the same time requires contextual information for disambiguation of meanings, for example. Mishra et al. (2020), for instance, propose different systems that tackle this issue. They present a collection of ML-based techniques to summarize scientific papers. Their solutions cover the possibility of generating different types of summaries with varying length; as well as lay summaries of technical papers – that is, summaries that don´t include any kind of technical jargon, but express their overall scope, goal, and impact. In a similar way, Rahul et al. (2020) proposes a solution that allows for two kinds of text summarization: extractive, i.e., creating a summary using sentences from the text; and abstractive, i.e., generating a new text that focuses on the key points of the original text.

One field which has found great use of NLP applications is healthcare. This has come in various forms. For example, Esteva et al. (2019) shows how ML-based systems can be used to transcribe voiced descriptions from a patient's symptoms into their medical records, liberating healthcare professionals from time consuming tasks. Similarly, Lu et al. (2021), analyses the efficiency of NLP techniques in identifying expressions of pain and fatigue by child and teenager cancer survivors, as opposed to the judgement of expert professionals. Most notably, NLP can be used efficiently in diagnosis processes. Hossain et al. (2021) shows how ML can be used to analyse the content of a person´s social media posts in search of signs of depression, which could help rise early alarms in the detection of one of the most prevalent causes of death in the world. Ong et al. (2020) surveys different ML-methods that have been proved able of, not only identifying strokes, but assessing their acuity and finding their location by analysing the patient´s speech. Also, through the analysis of a patient´s speech, Day et al. (2021) provides models to predict the severity and progression of aphasia. These predictions can be used by clinicians to plan and monitor the patient´s treatment. Interestingly, many of these examples show us the importance of some applications of STT techniques.

As further examples of ML usages in NLP and how broad their applications can be, we can have a look at Saxena (2022). This study introduces a methodology to enhance anonymity of personal data, providing a method to help safeguard the privacy of internet users. Osorio and Beltran (2020) present an application to detect activity of specific criminal cells – in this case, in Mexico –, which can proof of great utility to assist law enforcement agencies.

As promising and useful as ML can be for NLP, it comes with a cost. First, these algorithms rely on models, which are language and task specific. Those models can be complex and time consuming to create. Then, there is the computational (i.e., time and energy) cost for training the models. Kitaev et al. (2018) shows how fine-tuning a language model to serve a specific task in a specific language requires huge computational capacity. Fine-tuning is the process of using a model pretrained for a general task and training it further with fewer data to carry out the specific purpose of an application (Jurafsky and Martin, 2024).

Moving on to computational works on literature as a whole, and poetry in particular, we find some studies closely related to ours. Delmonte and Prati (2014) created SPARSAR. This tool analyses the style and prosody of a given to then infer an automatic, synthetic reading of the poem. Although the parting point and goal of that project are the opposite from ours, some of the techniques and lessons are useful to our task. Alm and Sproat (2005) did a similar job creating an automatic reader for fairy tales. But the most interesting work by Alm and Sproat for this thesis is that of Alm and Sproat (2005b). In this paper, the author studies the perception of emotions on people reading fairy tales aloud. This study analyses some of the metrics we are interested in: F0, intensity and speed (in rates of different granularities per minute). Of special interest are the papers and PhD thesis by Colonna (2020, 2021, 2024). In those publications she explores the prosody of readings of Spanish and Italians poets and conducts a set of analysis that may serve as a solid foundation for the study we propose in this thesis.

To finalise, let us explore the proposals that constitute the theoretical frame where this thesis fits in the context of literary criticism. As this is a computational linguistic thesis, we will not explore in depth the theoretical paradigm of literary criticism on which it is based. We will mainly base our work on the ideas proposed in Nowell-Smith (2015) and Mistrorigo (2018). They propose that poetry is not merely a form of written media, but that should be understood as a multimodal text that presents both a written text and a voiced text. Project PHONODIA, from which we will take our multimodal corpus, is a practical showcase of this proposal. This project is an archive where we can find poems by Spanish language modern poets alongside readings of those poems by their own authors. This means that we have access to the two formats that compose a poem. Since they consider poems as a multimodal text, this means that the structure of a poem is two folded: there is, on one hand, the structure of the written text, and on the other, the particular prosodic structure of a particular recording of a poem. Of course, different recordings taken at different times and locations may have different prosodic structures. This means that the structure of the voiced format is dynamic, same way as the written text can be revised in further editions.

The aim of this project is to create a tool that will facilitate a multimodal study of these archived poems. The tool will present an analysis of the voice of the reader, as well as concurrently showing how the structure of the voice relates to, and sometimes modifies or contradicts, the structure of the written text. It is important to remark that the analysis intents to highlight these relations between the two textual formats, rather than simply analysing the voice of the reader. What we aim to attain by this is richer analysis that what one obtains when focusing on just the written or the voiced halves the whole of the poem.

2. Notations

In this section we will provide an in-depth discussion of each of the three different notations – description, goal and how to read them – that shape the output of the program.

There are three notations, each displaying different information and useful for different research goals. The first of them displays the position of caesuras and enjambments. Secondly, we have a notation displaying information about changes in pitch, speed, and intensity of the recording of the poem provided. The third one also displays these changes in pitch, speed, and intensity, but through changes in colours, size, and font of the words, rather than in a scientific fashion, thought mainly for outreach.

2.1. Caesuras and enjambments

One of the main characteristics of poetry is its verse structure. A poem is usually formally conformed of different verses, be it in regular patterns or in a free verse structure. These verses induce a certain interpretation of how to read it: traditionally, the structure of the verses marks the prosodic structure of the voice that reads them. The end of a verse marks a pause in the reading, even if it may be unnatural to the reader due to the syntactic structure of the sentence. This notation will be accompanied by a graphic displaying the spectrogram and waveform of the recordings, as well as a spectrogram with both intensity and fundamental frequency highlighted, from which the presence or absence of pauses are drawn. As a quick summary, a pause is characterized by a lengthy drop in intensity.

As discussed, in the context of this work a poem is considered a multimodal piece. We consider it not only a written text, but a spoken one, too. This means that the prosodic structure uttered by the reader is a structural element of the poem itself, alongside the structure of the written text. If we consider the end of a verse as a sort of textual pause or break, it makes sense to consider that a pause or break in the reading rises this new structure that is built through the voice. In that sense, when the author reads the poem aloud, they may create “uttered verses”, i.e., a structure which divisions are pauses of the voice, that may differ from the written one.

This introduces two important terms. A caesura is a misalignment in the textual and verbal structures because of a break in the voice that is not reflected in the written verse structure. Coming from the written text, we can think of it as a breaking of the verse, or an unexpected pause. On the contrary, an enjambment occurs when an end of verse is not respected, and the reader continues reading uninterrupted the following verse. In other words, an enjambment is the absence of an expected pause. Both these concepts revolve around an unexpected event related to a pause, either because of its occurrence or because it was skipped.

These phenomena are the focus of the notation proposed in Mistrorigo (2018). The notation proposed in this thesis is developed from the one displayed in this book. The notation is presented, then expanded by successive examples of its usage, which we will use as an outline for the discussion.

With “Alto Jornal”, by Claudio Rodriguez, we are introduced to the symbols used to notate caesuras and enjambments. Pauses are represented with the symbol |, which may be, but not necessarily, a caesura. This is because a pause in the reading may indeed coincide with an end of verse in the written text. In this sense, the notation presents the new structure of the poem as a whole, marking every new verse independently of the written text. Further analysis by the researcher is needed to determine whether a | represents a caesura or not. Enjambments are represented with the symbol >. In the case of |, the symbol is used as an end of verse character, while > will always appear in verse. In the case of >, it will only appear between the last and first word of two verses of the written text, and only case of not pause being present between them in the voiced version.

This way the written structure is reshaped according to the voice of the reader. The previous, canonical, structure of the written text is displaced in favour of the verse structure that reflects the prosody of the reading. The following excerpt of the poem and the proposed transcription serves to illustrate it:

Dichoso el que un buen día sale humilde

y se va por la calle, como tantos

días más de su vida, y no lo espera

y, de pronto, ¿qué es esto?, mira a lo alto

y ve, pone el oído al mundo y oye,

[…]

Figure 1 Alto Jornal, fragment

Dichoso |

el que un buen día |

sale humilde |

y se va por la calle, |

como tantos > días más de su vida, |

y no lo espera |

y, de pronto, ¿qué es esto?, |

mira a lo alto > y ve, |

pone el oído al mundo y oye, |

[…]

Figure 2 Alto Jornal, transcription fragment

As we can see, all pauses have been marked with the symbol |, with independence on whether they represent or not a caesura. We can also see how a verse is now defined if, and only if, there is a pause. On the other hand, > represents a former end of verse (in the written text) that does not align with a pause in the audio, hence meaning that the two verses are joined together in one. In two other examples present in the book, two other marks are introduced a variation of this one. A double pipe, ||, to symbolize a larger than average pause, and /. The later symbolizes a pause long enough to be noticeable to the ear, but too short to justify classifying it as a caesura, and hence, to consider that it creates a new verse. It was decided that this last symbol would not be included in our notation because we want to focus on pauses with a structural significance.

Moving forward, we see the same exercise applied on “Siempre la claridad viene del cielo”, also by Claudio Rodríguez. Although the system applied remains the same, here a new element of the notation appears:

[…]
¿cómo voy a esperar nada del alba?
Y, sin embargo – esto es un don –, mi boca
espera, y mi alma espera, y tú me esperas,
ebria persecución, claridad sola
mortal como el abrazo de las hoces,
pero abrazo hasta el fin que nunca afloja.

Figure 3 Siempre la claridad viene del cielo, fragment

[…]
¿cómo voy a esperar nada del alba? > Y, sin embargo – esto es un don –, mi boca >
espera[,] y mi alma espera[,] y tú me esperas, > ebria persecución, |
claridad sola > mortal |
como el abrazo de las hoces, |
pero abrazo hasta el fin |
que nunca afloja. ||

Figure 4 Siempre la claridad viene del cielo, transcription fragment

As we can see in the transcription fragment, we have two commas encapsulated between square brackets ‘[,]’. This marks a comma that was skipped, i.e., a comma that is not aligned with a pause in the voice. We will further discuss this specific scenario in this chapter. Note that, in this example, || symbol simply marks the end of the poem and does not have, strictly speaking, the meaning defined before.

Now that we have laid the foundations of this notations as introduced in Mistrorigo (2018), let us explore the additions and modifications proposed in this project.

In Mistrorigo (2018) it was introduced the symbol | or (||) to indicate a pause. However, if we put a focus on annotating caesuras and enjambments, it would make sense to differentiate kinds of pauses. In this sense, we have two structural divisions that stem from the same perceptual phenomenon. That is, when we look at this analysis from a multimodal perspective and base our notation on both the written and uttered texts, it makes sense to distinguish between caesuras and pauses that match a written end of verse. This would enrich the notation and enhance its analytical and critical usage. Our proposal is to keep the | symbol to notate only caesuras and introduce the symbol ł to signify a pause in the voice that corresponds to a structural pause in the written text. This, however, leads naturally to the question of what represents a pause in a written text. This far we have only considered the end of verse as the written text representation of a pause. In written text we have punctuation symbols to mark, not only pauses, but also intonational features of the voice. It would make sense to consider the observation of these pauses when considering what represents a caesura. This discussion is especially interesting from the point of view of the enjambments; however, we must formalize what constitutes a caesura. In this case, we consider any pause in the voice as notation worthy. This would imply that any pause in the voice that does not correspond to either an end of verse or a punctuation symbol is caesura. This fully respects the understanding of a caesura as an unexpected pause (when basing our expectations on the written text). On the contrary, any pause that coincides with any of those textual markers is considered an expected pause and thus notated with the ł symbol.

We deem important to maintain the distinction of longer than average pause by doubling the corresponding symbol. This means, our symbols for pauses will be | and || for caesuras, and ł and łł for expected pauses.

The other elements being annotated by this notation are enjambments. As we previously said, and enjambment is the non-observation of a textual mark for a pause – usually, an end of verse. The way it modifies the written structure is by joining together verses, or part of verses, that appear in different lines in the written text. In figure 5 and 6 we can observe how there is and enjambment between the second half of the second verse and the first half of the third verse giving the following result:

[…] como tantos
días más de su vida, […]

Figure 5 Alto Jornal, fragment b

Turn into a single verse as shown in figure 6:

[…]
como tantos > días más de su vida
[…]

Figure 6 Alto Jornal, transcription fragment b

The annotating of enjambments naturally rises the same question we asked ourselves when discussing the caesuras. What about other pauses that are not end of verses? This mainly refers to quotation marks. As we can see in figure 4, whenever the pause introduced by a comma was not respected by the reader, the comma is enclosed between brackets '[,]'. However, this may not be totally clear for the reader of the transcription because we are introducing a new symbol with an intraverse usage. Furthermore, we are introducing a new intraverse symbol for a concept that already exists: a pause not being respected. Our proposal is to substitute the square brackets with <> for the sake of consistency with the symbol we are using for enjambments. In case where a verse ends with a punctuation mark and there is an enjambment, we propose both enclosing the punctuation mark and adding the > afterwards to show that neither the punctuation mark nor the end of verse where observed. In figure 7 we have the following example:

[…] y tú me esperas,
ebria persecución, […]

Figure 7 Siempre la claridad viene del cielo, fragment b

As seen in figure 4, there is an enjambment between these two verses. However, the first verse ends in a comma. Hence, the transcription as per our proposal would look like this:

[…]
y tú me esperas <,> > ebria persecución,
[…]

Figure 8 Siempre la claridad viene del cielo, transcription fragment b

This way, we indicate that neither of the textual marks for a pause have been observed in the reading. Of course, some of the commas in a text may reflect a grammatical usage that does not directly translate into a pause when read. However, this nuisance is highly dependent on various factors such as the sociolect or intent of the reader. Our decision then is to mark all of the punctuation marks that have not been observed and leave it up to the user to decide which ones are of special relevance for their case study.

To summarise, in this section we have presented a notation to mark the pauses and lack of expected pauses. This notation is carried out by modifying the written text to display the prosodic structure of the voice that reads the poem out loud. In this notation we use four different symbols:

- | to mark a caesura, and || to mark a longer caesura.
- ł to mark a pause that does not constitute a caesura, i.e., a pause that respects an existing textual mark, be it a punctuation mark or an end of verse. We consider them expected pauses but are anyway annotated

because they are a pause anyway and we prefer to leave it up to the user to decide its relevance in their analysis. As before, łł marks a longer expected pause.

- > to mark and enjambment.
- <> to mark that a punctuation mark has not been observed by the reader. For example, in figure 8: <,> >.

Let us look again at *Alto Jornal*, by Claudio Rodríguez. We have used extracts of this poem to illustrate the different aspects of this notation, so let us now use it as an example of the annotation with our proposed changes. As an example, we will annotate the first five verses of the poem. Here we will just provide the annotation as an example. For a discussing of the rationale behind the various decisions, see section 5. Analysis.

Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,

Figure 9 Alto Jornal, fragment c

Dichoso |

el que un buen día ||

sale humilde > y se va por la calle, ł

como tantos > días más de su vida, łł

y no lo espera > y, ł

de pronto łł

¿qué es esto?, łł

mira a lo alto > y ve, łł

pone el oído al mundo y oye, ł

Figure 10 Alto Jornal, annotated fragment c

In figure 10 we can see a full annotation of one the poems that we have been using as an example in this section, *Alto Jornal*, by Claudio Rodríguez. We can compare the new verse structure to the original one to have a clear picture of how the voice shapes it. Some of the pauses that have been notated as 'unexpected' can be considered, through syntactic or pragmatic arguments, that they are actually expected. However, we are only taking into account clear and conventional textual ways of representing a verse, which are and end of verse and a punctuation mark.

2.2. Paralinguistic features

In this section we will discuss our proposal for a notation that will include information of certain paralinguistic features. The features that we consider of special interest for an analysis of a poetic work are the variations in intensity, pitch and speed of the voice of the reader. To see other applications of these kind of analysis on the same theoretical framework proposed by Mistrorigo (2018), see

Colonna (2021; 2024). These three features are largely responsible for most of the perceptive phenomena of the human voice. In the context of a poetic reading, we can consider the sense of urgency by a sudden acceleration of the voice; or a feeling of sorrow of a segment that has a deeper and lower intensity voice that the rest of poem. Similarly to the notation of caesuras and enjambments, this notation will be accompanied by relevant graphics displaying the quantitative data used in the analysis. In this case, those graphics are a waveform, a spectrogram where intensity is highlighted, and a spectrogram where the fundamental frequency is highlighted.

We mean to display all three of these features in a single annotated text. This means that perhaps our biggest challenge is to find a balance between the richness of the notation and its lightness. The notation should be informative enough as to give proper value in the form of an informative insight of the annotated features. At the same time, it is important that the notations remain light enough so that the resulting text is easily readable for the researcher and their audience. To help achieve this, all the notations are drawn from previously existing and well-known notations, albeit not always from the current standard notations.

2.2.1. Intensity notation

First of all, let us explore the notation of the intensity variations. With this notation we aim at finding a visual display of the variations in the intensity of the reader´s voice. These intensity variations translate into a perceived variation on the volume of the uttering voice.

For this notation we have decided to make use of the standard notation for intensity variation of the musical language. This provides us not only with an already developed, spread and well-stablished notation, but also one that is informative enough as to allow musicians to properly understand the composer´s idea of how the piece should sound, and readable enough to be used together with plenty of other information in a sheet music – which, altogether, should be readable at first sight. This perfectly fits our purpose of finding a rich and informative notation that is designed to be used together with other forms of notated information. It is important to note that it displays variations of intensity and not absolute values of it. That means that these symbols will allow us to know that a specific segment is read at a higher or lower volume than the ones that surround it. This is our intended goal, as we consider that an absolute measure of intensity is not of interest for a prosodic analysis. It as well the intended purpose of the original notation, as musicians use other symbols to display information of absolute measures of intensity.

The symbols we will be using are < and > written below the corresponding segment of the text. They are shortened or elongated to match the written length of the segment they are marking. < means that the reader is gradually increasing the volume of their voice for the duration of the segment. >, on the other hand, means that the reader is gradually decreasing the volume of their voice. There is another usage for these symbols, although is not always used this way by all authors. That is using them under a single note – in our case, a single syllable – , to mark a sudden and very short variation. This would be a syllable that is uttered at a notably higher or lower volume than the rest of the word.

Let us see some examples of its original usage. In figure 11 we can see an example of a *crescendo*. That is the musical term that refers to a gradual increase in intensity of the notes being played.

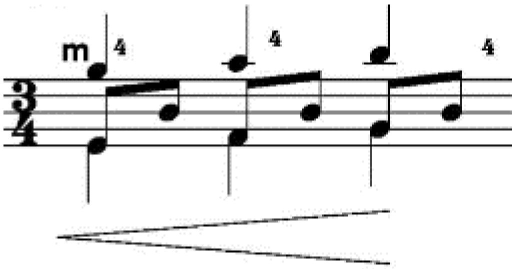


Figure 11 Crescendo (source Lágrima, Francesc Tàrrega)

In the example above, the span of the *crescendo* is of five notes. As we explained above, this is not a mark of absolute intensity, but rather of variation. In this example, the first note (leftmost black symbol in the pentagram) is played at a lower volume. From there, the volume keeps on increasing, each note being at a higher volume than the previous one, until the last note (rightmost black symbol in the pentagram), where it reaches its peak.

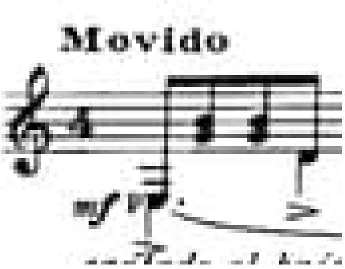


Figure 12 Accents (source 20 Estudios Sencillos, Leo Brower)

In figure 12 we can observe two examples of accents. An accent is a local intensity peak. It is represented by the same symbol as the *crescendo* but spans only one note. It means that this note is played at a higher volume than the previous and next notes. One can observe it under the first and last note.

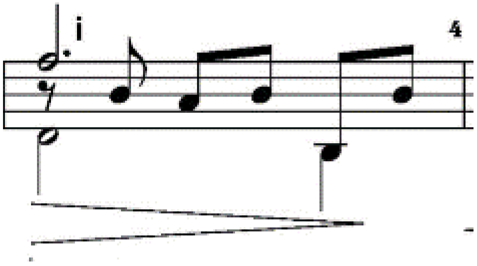


Figure 13 Diminuendo (source Lágrima, Francesc Tàrrega)

In figure 13 we see a *diminuendo*. This can be understood as the opposite of a *crescendo*. The first note is played at a higher volume than the next one, and each one gets increasingly softer than the previous, reaching its minimum at the last note.

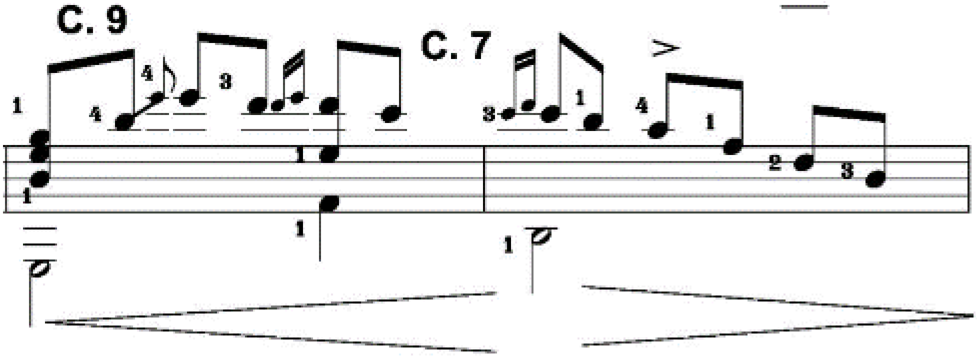
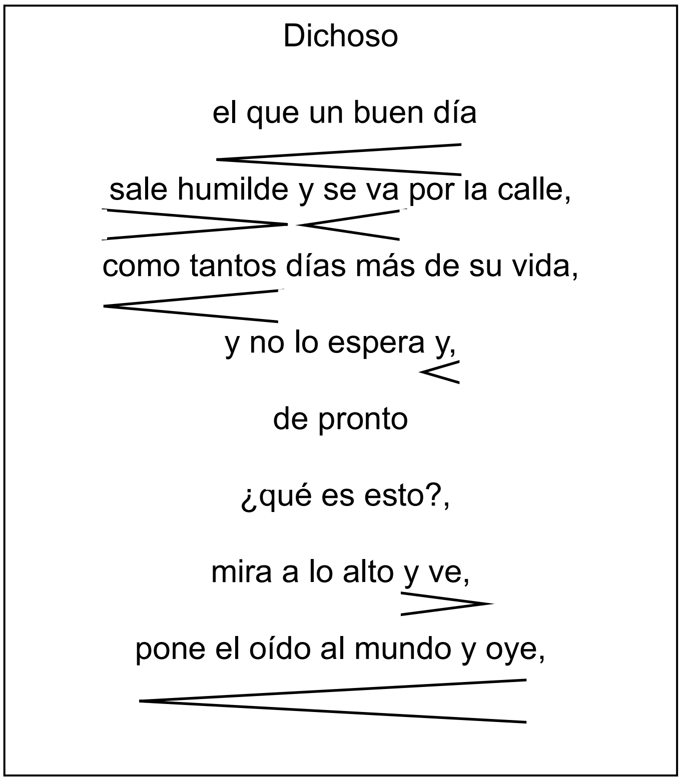


Figure 14 Crescendo & diminuendo (source Adelita, Francesc Tàrrega)

As a last example, in figure 14 we can observe a *crescendo* immediately followed by a *diminuendo.* Both span two notes. This means that volume increases from the first note of the *crescendo* to the second one and diminishes from the first note of the *diminuendo* to the second one. First and last note of the image are not affected by either the *crescendo* nor the *diminuendo.*

Let us see what this notation looks like applied to the first five verses of *Alto Jornal*. Note that this annotation has not been carried out by an expert phonetician. It should be understood as an example and not as a definitive version. As discussed further in the text, the involvement of experts trained in this kind of prosodic analysis will be needed to polish both the analysis and the result.



Dichoso

el que un buen día

sale humilde y se va por la calle,

como tantos días más de su vida,

y no lo espera y,

de pronto

¿qué es esto?,

mira a lo alto y ve,

pone el oído al mundo y oye,

Figure 15 Alto Jornal, intensity annotation

2.2.2. Pitch notation

In this section we will discuss our proposal for a notation system for the pitch and intonation features of the recording.

As with the previously discussed notation for intensity variations, we want to find a notation light enough to be integrated in the text alongside two other notations, while still being informative and of interest and value to the user. Among the different notation systems that are currently used in phonetics research, we have decided to settle for the ToBI system (Silverman et al., 1992) (Beckman et al., 2005). In particular, we will be using the Sp_ToBI system (Aguilar et al., 2024), a modification of the original one – which was developed to serve specifically as a notation system for American English – presented in Beckman et al. (2002). Sp_ToBI has been further developed and modified. In particular, we will be using the system documented in the web page of the Pompeu Fabra University (Aguilar et al., 2024), which includes the modifications proposed by Face and Prieto (2007), Estebas-Vilaplana and Prieto (2009) and Estebas-Vilaplana and Prieto (2010).

The labelling system of Sp_ToBI is divided into five tiers. From those, we only consider the fourth tier – the tone tier – for our notation system. We won´t consider the fifth one – the miscellaneous tier – even if it could be of interest for our purpose, due to its lack of standardization. This tier proposes labels for pitch accents and boundary tones. Pitch accent is a concept that refers to a syllable that is prominent among the surrounding ones due to a contrast on its relative pitch when compared to those that surround it. In Spanish, that can take two different forms: it can either be a high (H) or low (L). Thus, this is a concept that relates directly to how a syllable is perceived to stand out among its neighbours. At first glance, one could consider this concept to be the one of most interest when studying the prosody of poetics, because we generally associate the study of poetics to the study of syllables through concepts like metrics. However, given the nature of our study, which is an extension of a study of caesuras and enjambments, we decided to focus on boundary tones. Boundary tones mark the beginning and end of prosodic phrases. This feature allows us to empirically mark perceived tonal differences that have a great communicative significance. For example, boundary tones allow speakers to differentiate a question from a statement; to wait for a next item in an enumeration or understand that it has finished, etc. Hence, we can extend our dichotomy of the expected and unexpected by finding boundary tones. In this context, we can detect, for instance, a boundary tone that marks a question when there is no question mark in the text, or vice versa. This notation can also provide valuable insight into the understanding of the poem itself, by identifying prosodic queues that do not match our expectations, or that give weight to an interpretation where ambiguities may be found.

These boundaries can be of two types: intonational phrases (IPs) and intermediate phrases (ips). IPs mark the end of major prosodic units, within which minor units may occur. These minor units are ips. Boundary tones in Sp_ToBI are not accompanied by pitch accents as in the original ToBI system. This is because there has not been found evidence that supports their necessity to account for all tonal movements. Another remarkable difference between ToBI and Sp_ToBI is that Sp_ToBI introduces an intermediate tone M. It is found in certain enumerations and questions, and when it appears as an IP (M%) of a statement, it marks a hesitation. In table 1 we can see a summary of all IPs, and of all ips in table 2, in the Spanish language, as described in the Sp_ToBI:

Boundary Tone	**Description**	**Appearance**
L%	Low sustained tone or a low descending tone	At the end of both broad and narrow statements, imperatives, anti-expectative and imperative yes-no questions
M%	Falling movement to a mid tone target, or mid level plateau after a high tone	At pedagogic enumerations, hesitation statements, polite yes-no questions and stylized vocatives
HH%	Very sharp rising pitch	At the end of yes-no questions
LH%	Dip and then a rise to a high F0	At anti-expectative and invitation questions
LM%	Dip and then a rise to a mid F0	At obviousness statements

HL%	Rise and then a fall to a low F0	At (after a high or low pitch) exhortative requests, emphatic exclamatives and insistent vocatives
LHL%	Fall, then rise and then fall to a low F0 value	Exhortative requests

Table 1 IPs in Sp_ToBI

Boundary Tone	Description	Appearance
L-	Low sustained tone or a low descending tone	After left-dislocated elements in broad focus statements, before a right-dislocated statement in broad and narrow focus statements, imperatives, falling yes-no questions, wh-questions, etc.
M-	Falling movement to a mid tone target or a mid level plateau after a high tone	Pedagogic enumerations and at the end of initial elements of questions
H-	Rising pitch movement, coming from either a high or low pitch accent	At the end of non-final constituents, inconclusive statements, etc.
HH-	Very sharp rise	At the first part of alternative polar questions

LH-	Dip and then a rise to a mid F0	At anti-expectational and incredulity questions and in obviousness statements
HL-	Rise and then a fall to a low F0	At exhortative requests
LHL-	Fall, then rise and then fall to a low F0 value	At exhortative requests

Table 2 ips in Sp_ToBI

In the following figure we present an example of the firsts five verses of *Alto Jornal*, annotated with the tags from Sp_ToBI. As already explained in the previous section, this is just an example of the use of the notation. This has to be understood as a showcase of the proposed system, and not as an accurate analysis of the pitch of the reader´s voice.

Dichoso
M%
el que un buen día
L- HL%
sale humilde y se va por la calle,
L- L%
como tantos días más de su vida,
H- H%
y no lo espera y,
HH%
de pronto
HH%
¿qué es esto?,
M- L%
mira a lo alto y ve,
H- H%
pone el oído al mundo y oye,
H- LHL%

Figure 16 Alto Jornal, pitch annotation

2.2.3. Speed notation

In this section we will discuss a notation for speed variations in the poets´ voice during the readings. This notation is perhaps the hardest to detail, as speed measurement in prosody studies is a topic that is very much still open for discussion, as well as its interpretation and perception by the listener. Henceforth, we will refer to it as “speed tempo”, which is the term most commonly used in the literature.

Speech tempo, as a measure of speed, is defined as the number of speech units of a given type per unit of time. One of the challenges when analysing it comes from the discrepancy between measurement and perception. While all other previously discussed phenomena can be reliable judge based on perception, that is not the case with tempo. Leendert et. al (2022) show that fundamental frequency and intensity have an effect on perceived tempo. Koreman (2006) shows that clarity of utterance also has an impact on perceive tempo. These studies prove a misalignment between tempo an empirically measurable phenomenon and tempo as a perceived phenomenon. Notating tempo then will always come as a delicate matter, as any analysis resulting from the notation must be a compromise between measured and perceived tempo. This is because all other features have a clear relation between the perceived and measured phenomena, so it makes sense to use those features for studies on the perception of poetry, from which depends the understanding that the listener draws from the poem and the effect that the recital as an artistic performance causes on them. It is important to stick to the context from which we part, that is, this is a study on poetry, so dismissing its artistic layer to reduce the whole to a mathematical abstraction – i.e., considering only a measurement that does not align with perception – would be of little interest, if not a mistake, no matter how much it could simplify the work.

Usually, the time unit are seconds given the unstable nature of speech, and because the usual units of speech fit well within this measure of time. When faced with the decision of which linguistic unit to use for the measurement, the two obvious options are syllables and phonemes. Both these options present a number of advantages and disadvantages each.

Let us first examine the use of syllables per second as a unit of speech speed. At first, it may see, like the more natural option for a study on poetry. After all, syllables are often used as the base unit for study poetry, especially written poetry. Concepts like metrics, rhythm and certain kinds of rhymes stem from the concept of syllable. However, if we intend for a multimodal analysis that seeks to exploit what each modality has to offer, focusing on something that serves as a base for most studies on written poetry may not be desirable. This is because we may find ourselves tied by concepts that were originally developed and use for a unimodal analysis, hence severing what multimodality offers. Second, there are both phonetic and computational arguments to be made against the use of syllables per second. Uttered syllables may not correspond to written syllables if one uses the standard to determine how many syllables there are. In the Spanish language there are, for instance, two tendencies in certain accents and registers that may yield fewer uttered syllables than expected. Take the sentence "¿Qué ha pasado?" – trad. "What happened?". According to the standard, one would expect the following transcription: [‘kea pa’sa.ðo]. However, it would not be strange in peninsular Spanish to actually hear [‘ka pa’sao]. This means there may not be an accurate correspondence between the written and uttered syllables. This would still make for an interesting study, as the above presented alternative transcription can be considered heavily marked, and hence be an intentional decision by the author to pronounce it in that specific way. Another aspect to consider, is that syllables can be of various lengths – from one sound up to four. That means that an array of shorter syllables would appear as uttered faster than an array of longer ones. This may not correspond to the perceived speed and hence be confusing or misleading when. On a computational level, syllables are a complex structure because they are formed by different sound with no clear boundary. Typically, when working with syllables a "dictionary of syllables" is created. That is, a list of all the possible – or all the expected – syllables that we can encounter. Then, the analysis is performed by matching each syllable to one on that list. This creates a problem, as there is no accurate way of processing an unknown syllable, i.e., one that is not on the list – for example, a word from a different language, or a syllable with a very rare occurrence within the language. Typically, these cases are solved by creating an “unknown” tag to classify everything that does not match a syllable from the list. This is not ideal because of varying duration of syllables that we have previously discussed.

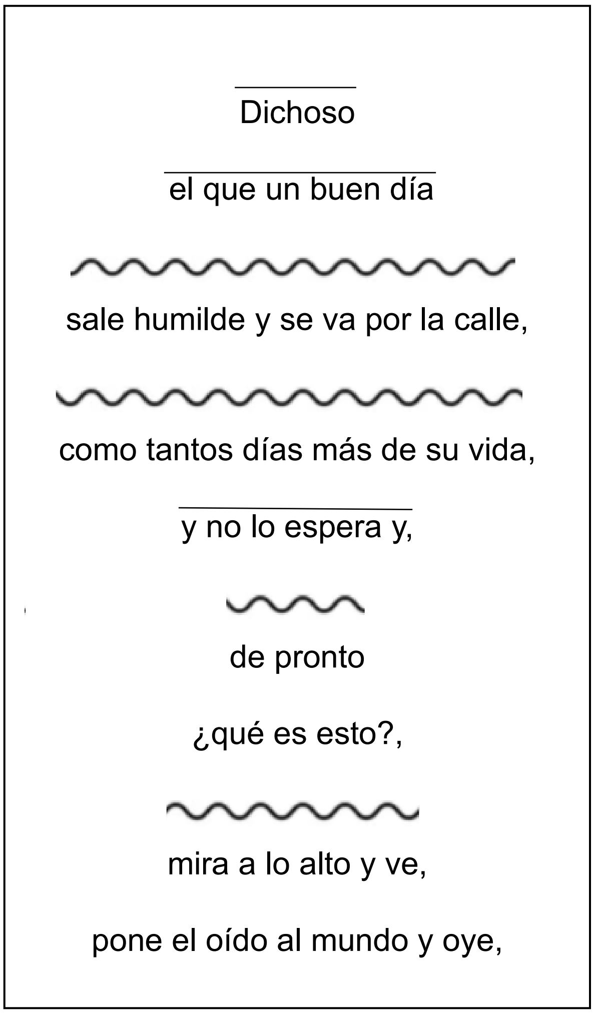
The other option is the use of phonemes. This option is computationally easier and more precise, as a phone is more clearly defined, both what constitutes one, and where it starts and ends. However, there are other problems. Considering the same sentence considered above, a faster speech may be achieved by eliminating sounds. This could lead to a situation where the fastest speech has less sounds per second because it has less sounds as a whole. Due to its computational simplicity and to having less variation at a statistical level, this is the option we have opted for.

Lastly, it must be discussed whether to take pauses into consideration when it comes to calculating the speech tempo. If we accept that pauses function as a prosodic boundary, do they pertain to a prosodic phrase? Given that pauses are already taken into consideration to restructure the poem around them, the fragments where the speed measurement is of interest already falls within pauses. Because of this, our proposal is to calculate speed variations only within pauses and compare those against their neighbours.

Given the difficulties to find an adequate way of defining and measuring the speech tempo, the discrepancies between perception and measurement, and the lack of a standard to annotate it, we will present a barebones proposal. We think that finding a suitable notation system for speech tempo is something that requires insight from many fields that will only come with the use of the tool. These insights can lead to a number of iterations of the development of the notations until we can reach one that fulfils the needs of the research community.

Our proposal, hence, is rather simplistic. We have opted for the use of only two symbols, that will be notated above the corresponding fragments. A straight line – above a given fragment indicates a lesser speed than the average of the poem. An undulating line~represents a fragment that is uttered a higher speed than the average.

The next figure is an example for this notation.



Dichoso

el que un buen día

sale humilde y se va por la calle,

como tantos días más de su vida,

y no lo espera y,

de pronto

¿qué es esto?,

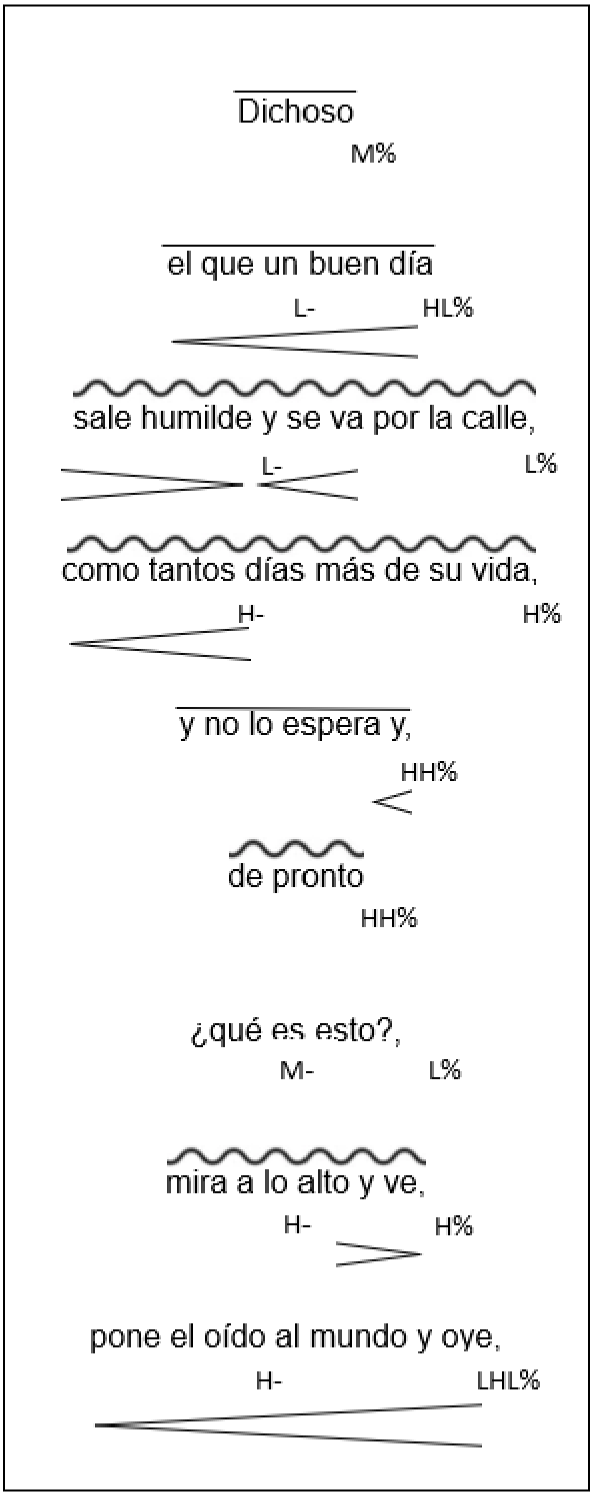
mira a lo alto y ve,

pone el oído al mundo y oye,

Figure 17 Alto Jornal, speed annotation

2.2.4. Example

The intention for the phonetic notation is to display the information conveyed in the tree above subsections. For this reason, it will be presented to the user as a single annotated text instead of three. In the next figure we can see an example of the combined notations into a single text. We will use the first five verses of *Alto Jornal* as with the previous examples.

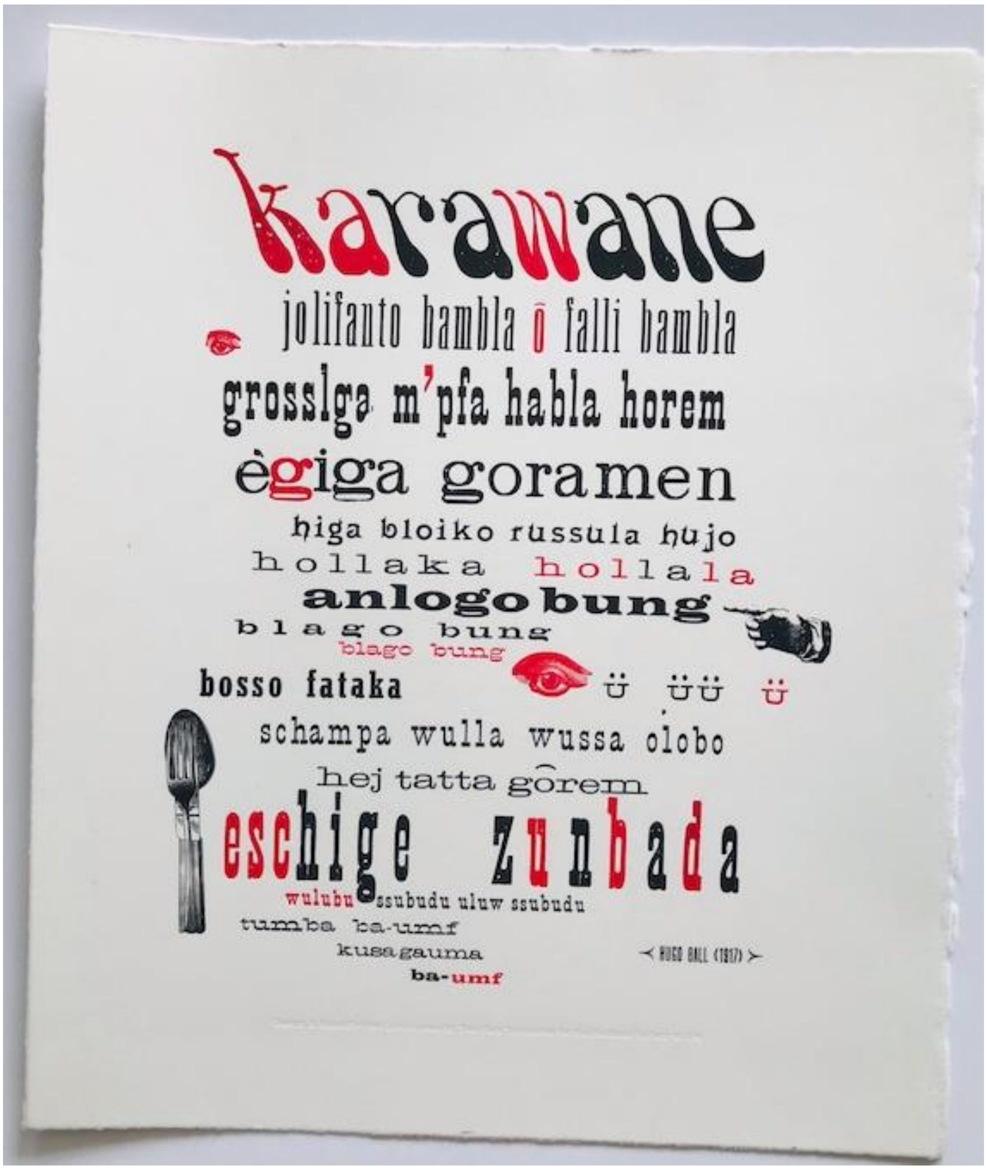


Dichoso
el que un buen día
sale humilde y se va por la calle,
como tantos días más de su vida,
y no lo espera y,
de pronto
¿qué es esto?,
mira a lo alto y ve,
pone el oído al mundo y oye,

Figure 18 Phonetic annotation example

2.3. Optophonetic notation

In this section we will discuss the last of the notations. The intent of this notation is to allow for a visually impactful restructure of the text. Its purpose would mostly be outreach as it will not display information with the usual symbolism of scientific notations. Rather, the intent is to use the previous analysis to alter the text as to transmit through the text what the poets achieve with their voices. Although the scheme to construct the output text is as empirical and data driven as the other notations, the output is closer to an artistic representation of the poets' voices. The inspiration for it comes from optophonetic or optophonic poems, like "Karawane", by dadaist artist Hugo Ball:



karawane
jolifanto bambla ô falli bambla
grossiga m'pfa habla horem
ègiga goramen
higo bloiko russula hujo
hollaka hollala
anlogo bung
blago bung
blago bung
bosso fataka
ü üü ü
schampa wulla wussa ólobo
hej tatta gôrem
eschige zunbada
wulubu ssubudu uluw ssubudu
tumba ba-umf
kusagauma
ba-umf

Figure 19 Karawane, by Hugo Ball

This poem uses invented words that bare no semantical meaning or structure whatsoever. The actual meaning of the poem comes, first from its nonsensical words themselves, but also from the use of colour, size, font, etc. to shape the words into a structure that evoke sound. In a similar manner, our intent is to find a system to reshape any given poem accompanied by a transcription into a visual representation similar to that of Karawane. That is, colouring, resizing, altering the font, using bolds or italics, etc., to represent variations in intensity, speech and speed. The goal is to achieve a representation that evokes the voice of the poet.

Despite this being the final goal of for this notation, in this project we will restrict it to an intermediate step. We will develop a notation system that will mark the elements that have to be altered by the future, final notation. This final notation will be discussed as a future work.

Because this is not a final result, but rather an intermediate step that will serve as basic guide for the final notation, it consists of the minimal components needed to represent the features that we want to see reflected in the final representation. Because of this, this notation is not as rich as the others. The goal is to indicate which changes will be applied to each word or fragment, so the indications are more simplistic in essence than those of the previous notations that intended to be more exhaustive.

Our inspiration for this notation is again found in the extremely rich musical language. In this case, we propose the use of a tetragram – like, for example, the ones used to write Gregorian chant – to be placed over the verse. Each line of the tetragram will be used to annotate a different feature of the ones we annotated with the previous notations. These features are, on one hand, those related to the interpretative performance: caesuras and enjambments. On the other hand, we will annotate in a simplified manner the three prosodic features of the previous sections: pitch, intensity and tempo.

We propose a different restructuration of the verses for this notation. Because the shape of the final optophonetic representation is yet to be worked on, we propose a structure that we believe will bring more flexibility. Instead of breaking the verses at every caesura, we will respect the integrity of a verse as written. At the same time, we will join together verses where an enjambment occurs. What we are doing, then, is respect the full length of verses in both their written and uttered forms and taken that allows for greater structural continuity.

The first line of the tetragram will annotate caesuras and enjambments. It will consist of the straight line of the tetragram, interrupted only by the symbols ';' – short break – and '.' – long break. This means that enjambments will not be represented per se, but for the fact that the line remains uninterrupted where an end of verse appears in the written poem. For this reason, and because we will merge verses that have an enjambment between them, we will marc the enjambment with the symbol used in the previous annotation: >. We will use, though, only in the case of continuity between verses and will not annotate non-observed pauses.

The second line of the pentagram will represent the speed variations. To represent them we have chosen the symbols << >> to be used similarly to brackets. They will encapsulate fragments where there is an acceleration or a slow down of the reader's voice. Because of their usage and for the sake of clarity, we will call them brackets from now on. A fragment that is uttered at a lower-than-average speed will be annotated beginning with the left-pointing brackets << and will be closed with the right-pointing brackets >>. The opposite will be true for fragments that at uttered at a greater-than-average speed: it will be encapsulated beginning with right-pointing brackets >> and will be closed by left-pointing brackets <<. The purpose of this is to mark the point on which the voice starts to accelerate and to slow down again, or vice versa.

The third line is reserved for annotating the pitch variations. Again, for the sake of simplifications we will just mark outstanding high and low points. Otherwise, we believe that the notation would become overcluttered and would make for a different transformation into the final visual form. We have to bear in mind that this annotation is intended as an intermediate step for a visual interpretation of the poem that will, in a way, evoke the phonetic aspects that we are annotating now. The annotation will consist of slash bars in an arrow shape, pointing up or down depending on whether the frequency goes up or down in the recording: /\ for a high, \/ for a low. We believe this is a simple, easy to understand way of annotating exactly what we want to point out, which is some relevant maximums and minimums of pitch.

Lastly, the fourth line will be used to display changes in intensity variations. In order to maintain consistency, we will use a simple notation, easy to understand, that can be placed just over the specific words that we want to highlight. The symbol o represent it will be the exclamation mark "!" for low intensity points and an inverted exclamation mark "¡" for high intensity points. These symbols can be placed over the words where this highs and lows occur, and over as many consecutive words as necessary, in case the high or low intensity is maintained over a period of time.

```
        < ______________________>>___________________________<
         ______________\/________/\\/__________\/_________________\/
          _________!_______________¡__________!________¡__________
          ________;________________.______________________________;
      Dichoso el que un buen día sale humilde y se va por la calle,
               >_____________________________><______________>
           ___________/\___________________/\______________/\
               !___________¡_________________________________¡
               ____________________________.______________;
             como tantos días más de su vida, y no lo espera y,
>_____<_____________>__________________<___________________________
_____/\/\___________\/___________/\_______/\___________/\____________\/\/\/
 ____________________________________¡__!___________________________¡
  _________._____________._______________.__________________________;
de pronto, ¿qué es esto?, mira a lo alto y ve, pone el oído al mundo y oye,
```

Figure 20 Alto Jornal, optophonetic notation

3. Tool

In this section we will discuss the tool itself, as well as going through the interface of the tool to get an overview of how it works, and of the function of every component. Because the program is not fully developed, its current version is a proof of concept, and some the functionalities here explained are yet to be implemented.

This program consists of an interface developed with the PyQT library, and a backend developed in Python. Treatment of the raw input data is done partly through the use of the Parselmouth (Jadoul et al., 2018) library, a public API for Praat (Boersma, 2001). The rest of the work, which will be discussed in more detail in section “Analysis” is done through the use of common Python libraries for data processing and plotting.

The code is, as per the writing of this thesis, not publicly available at any repository. If the reader wishes to have a copy of it for their own use, do not hesitate to write an email to 893604@stud.unive.it. As for the time of the writing, the software is delivered as its source code, and no installation is needed. This means that user need have a mean to run the code through a Python interpreter to execute it. Typically, a Python interpreter comes installed by default in any computer with a Windows or iOS operative system. In this case, Python 3.9 is needed for running the program, and it is not guaranteed that it will function normally, or even at all, with other versions of Python. To ease the first use, there is within the project an executable file with the external – software – requirements to run the project. Those external requirements consist of public python libraries. Again, should the user need any help setting up the environment to run the code, do not hesitate to contact the author at the above email address. The lightness of the program in its current form allows for a high portability, as it fits without any problem in a flash drive and makes it so that no installation is needed. This should not change in any foreseeable future.

Once we run the program, we are met with what we could call the “Home Screen” of PAUSE. It consists of a drop-down menu and four panels. The menu will display the different utilities of the tool. As per the panels, the left-most panel, which is the biggest, will display the input text. It also contains a slider that will be used to control the listening of the input audio. Top-most panel of those to the right will contain the plots of the different features analysed by the tool. Those plots are always scrollable and zoomable, and they display precise numerical information with the cursor is left over any of the plotted points. In the case of combined plots, zooming into, or scrolling onto one of them mimics the action on the other plot. This means that the information that the user sees is always concordant between the two plots. Bottom-left quarter of the screen is divided into two panels. The on the left will display the output of the analysis, which consists of the annotated text. The one to the right will display the legend of the notation.

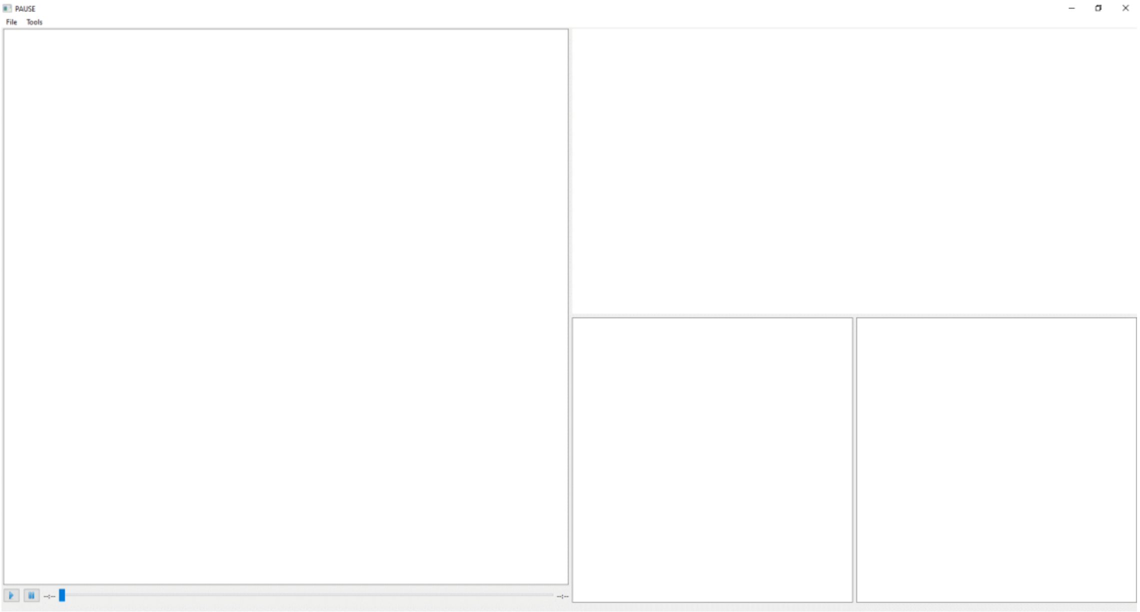


Figure 21 Home Screen

Now let us examine the first of the drop-down menus. It is called “File” and it contains the basic set of actions needed to interact with the tool. It allows to load text and audio files to be processed, as well as saving the results in various formats. It will also allow the user to lad a previous project to resume the work. We will now briefly cover all the options.

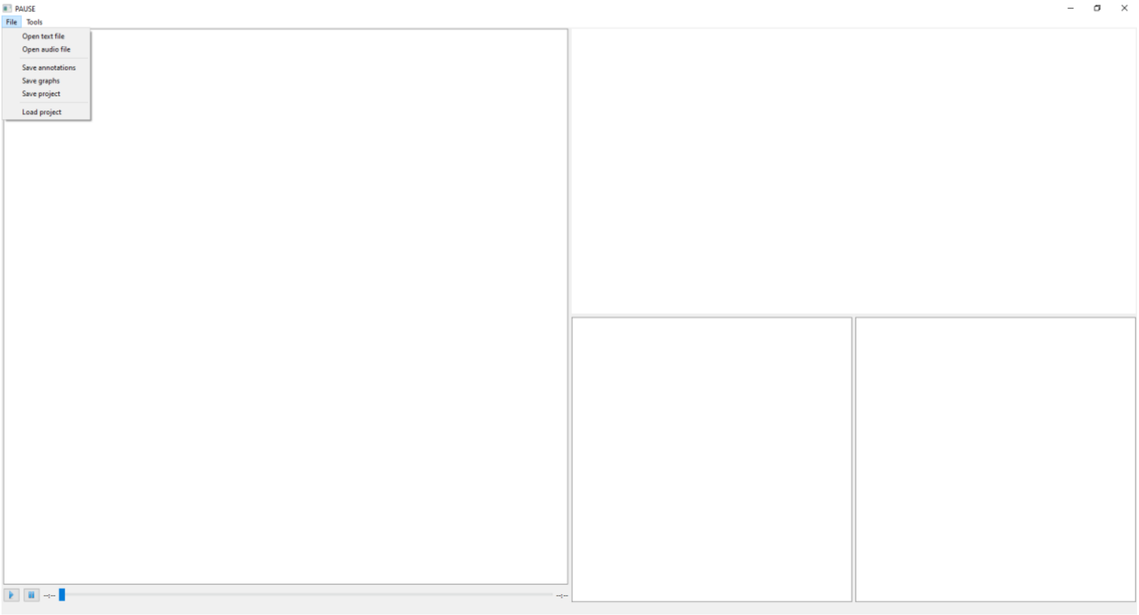


Figure 22 File menu

First option we see is “Open text file”. As the name indicates, it allows the user to select and open a text file, which will be displayed in the corresponding panel. It works by opening a browsing window set at the project´s folder. This window, by default, will only show files with the .txt extension.

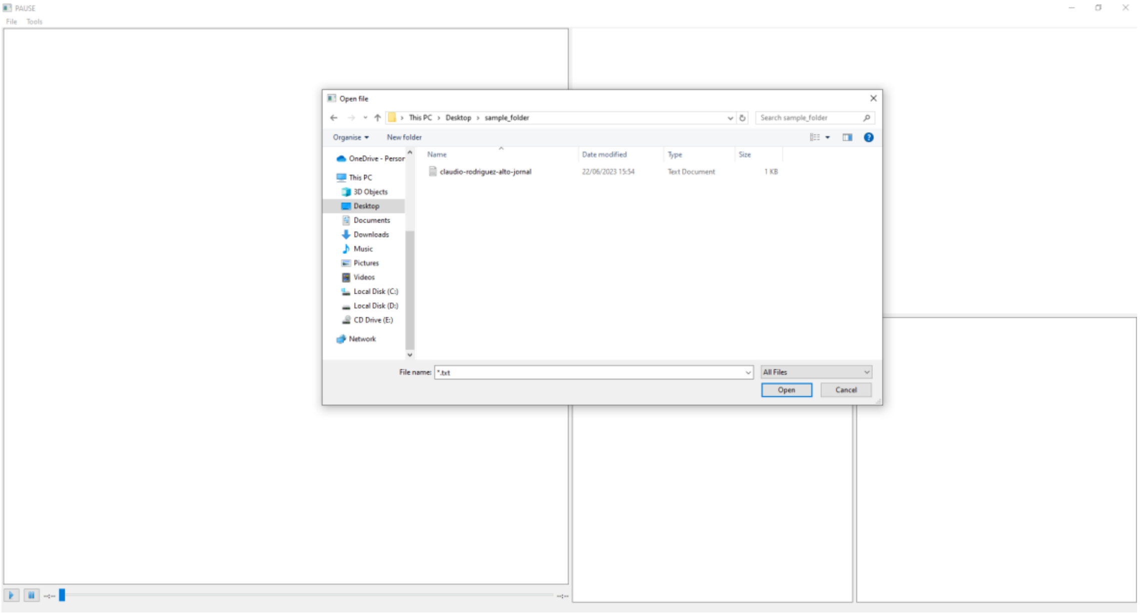
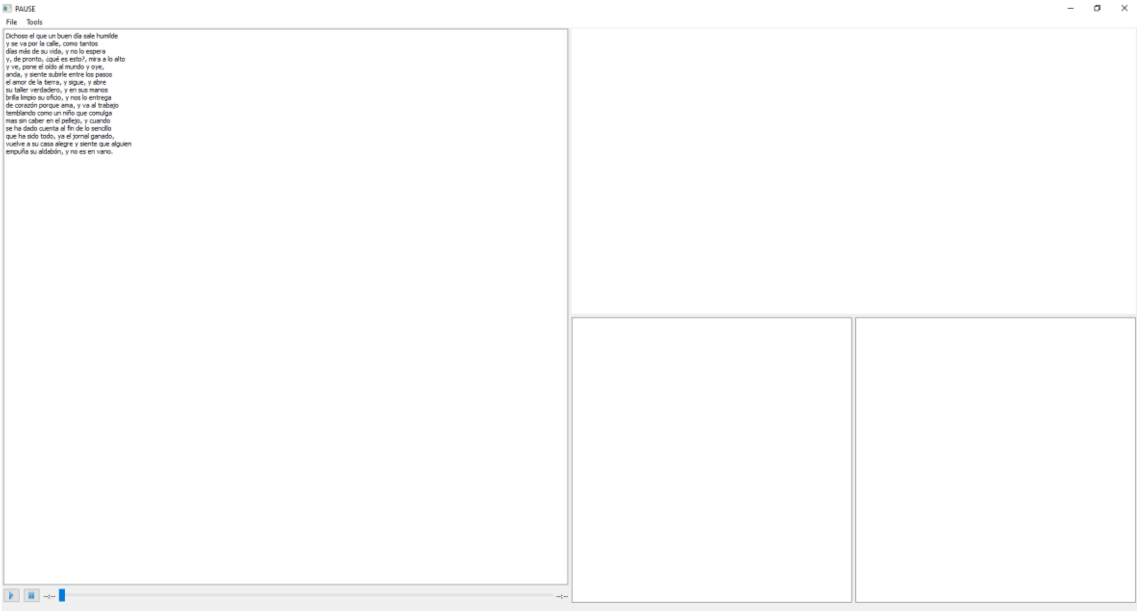


Figure 23 Text file browser

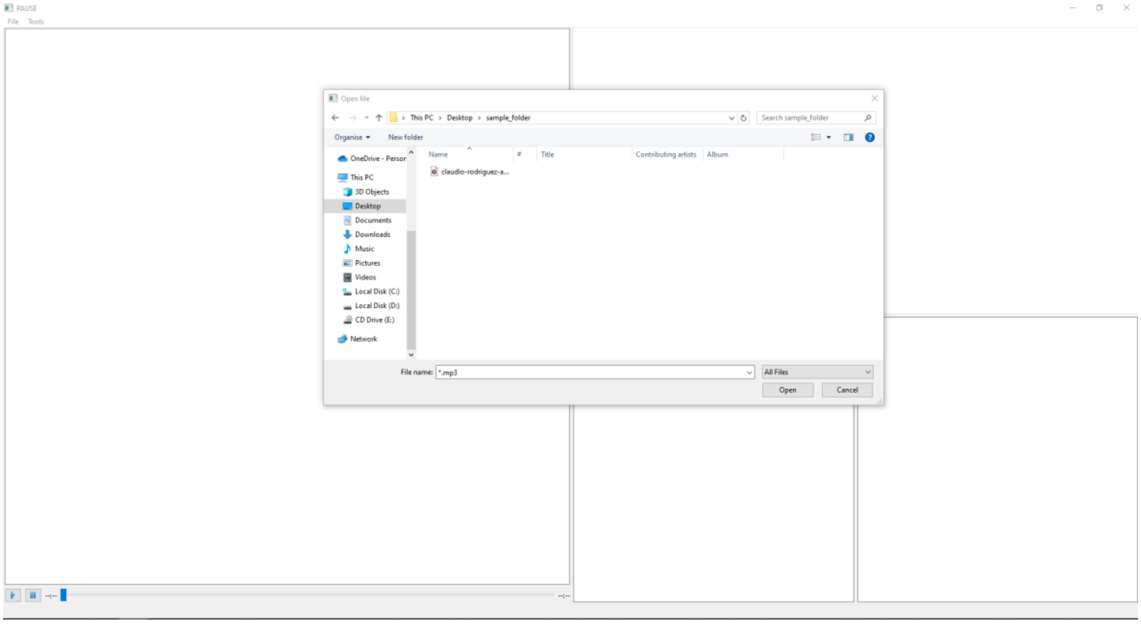
Notice in the next figure how the text within the file we selected is displayed in the left-most panel:



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 24 Text file loaded

In a similar manner, the “Open audio file” will again open a window to brows for an audio file. In this case the window is set by default to only show .mp3 files.

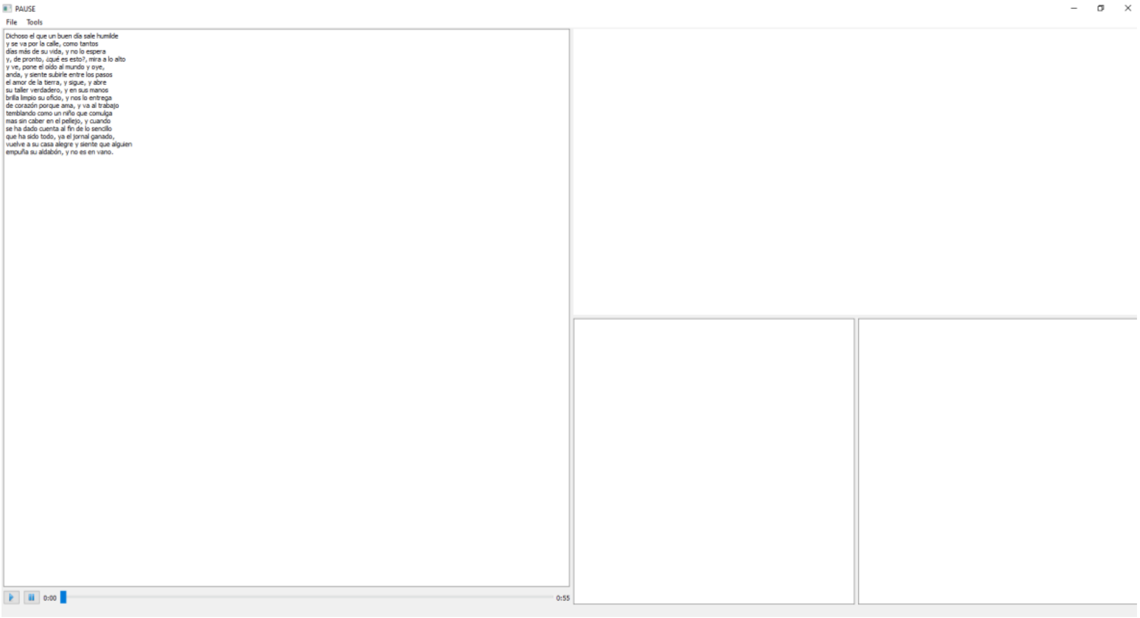


File name: *.mp3

All Files

Figure 25 Audio file browser

In the next figure we can see both text and audio loaded. We can make sure that the audio has been properly loaded by observing the slider. It should now display the duration of the recording. We can as well press play to listen to the audio.



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 26 Text and audio files loaded

Before moving forward on the “File” menu, let us explore the “Tools” menu. Currently, this menu consists of just one option, another menu called “Analysis”, they are placed like this so that the “Tools” menu can serve as a placeholder for future utilities of the program. The “Analysis” menu is also drop-down, and in this case, it will give access to the analytic functionalities of the program. The three options displayed, “Caesuras and enjambments”, “Phonetic” and “Optophonic” correspond to the three notations detailed in the previous section. Note that “Phonetic” corresponds to combination of the notations explained in the subsection *Paralinguistic features*. Once selected, relevant plots will be displayed on the top-right panel, and the output and the necessary legend will appear on the one below it.

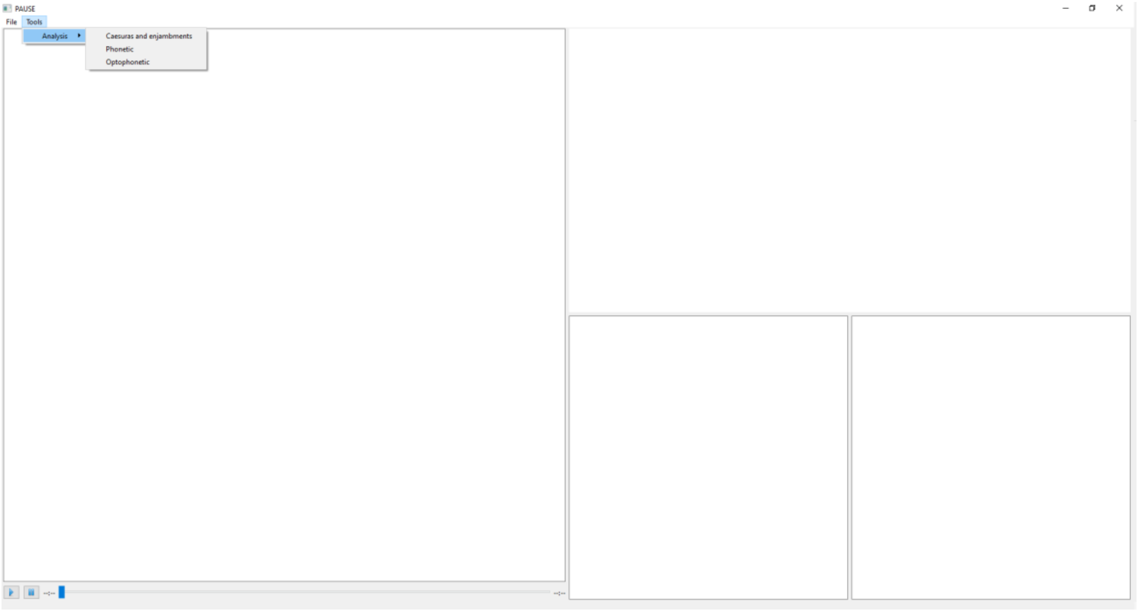
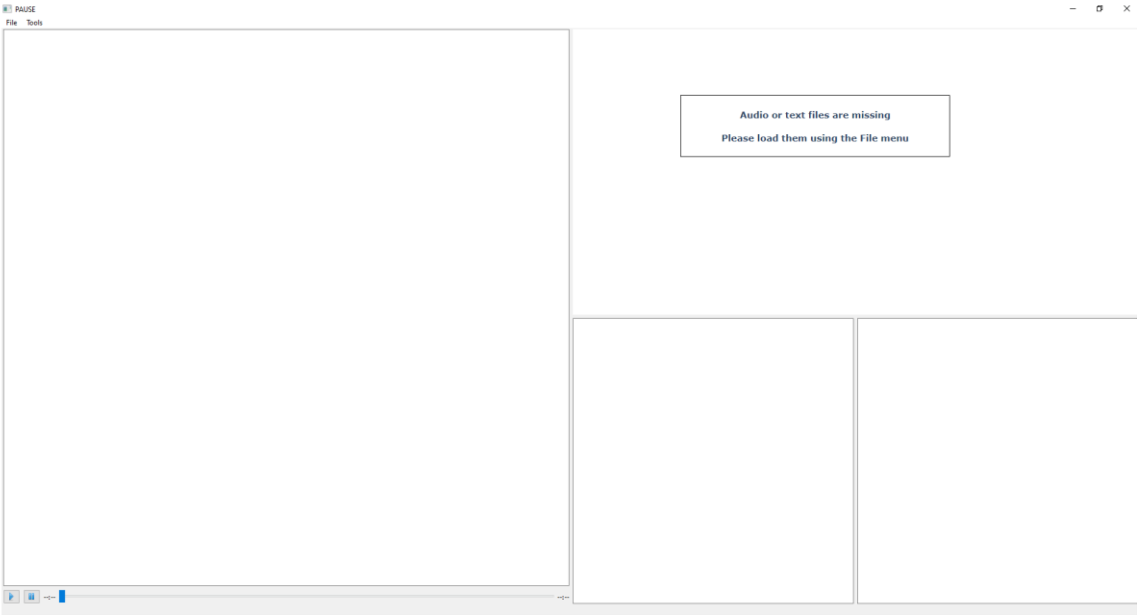


Figure 27 Tools menu

As already explained, each of the options in the “Tools/Analysis” menu will run the corresponding analysis. However, let us first imagine the scenario where the user forgot to load the text or audio files – or both – before selecting one the analysis. In this case, of course, there is nothing to be analysed, and so the user will be prompted to first load the necessary files.

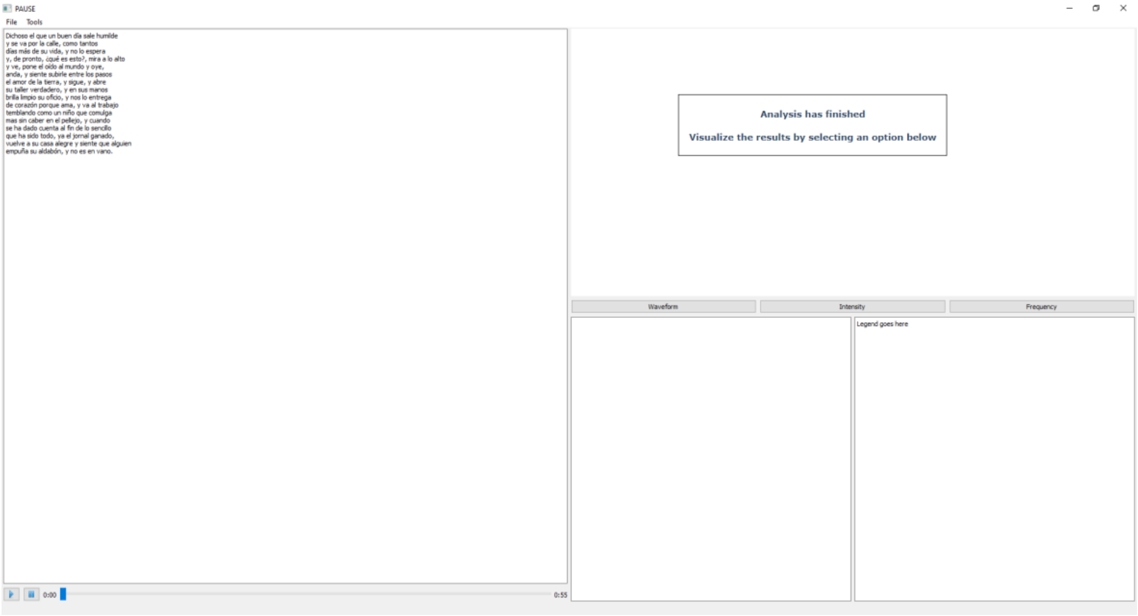


Audio or text files are missing

Please load them using the File menu

Figure 28 Prompt to load files

If the user now selects the “Caesuras and enjambments” option, they will see the spectrogram of the audio on the top-right panel. We will see examples of this in the Use Cases section. However, if they select any of the options, which are supported by more than one plot, the user will receive a message indicating that the analysis has finalised and to select which plot they desire to visualise. In the next figure, we can see how three buttons have appeared under the plot panel. Clicking on a button will display the requested plot. The pressed button will remain greyed out as a visual cue to facilitate knowing which plot the user is seeing. Clicking on any other button will change the plot in display. In the case of the paralinguistic features notation, it will also change the legend to match that of the feature being observed in the plot. This is because the legend would otherwise be too big, and we believe this will simplify the lecture of the annotated text.



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

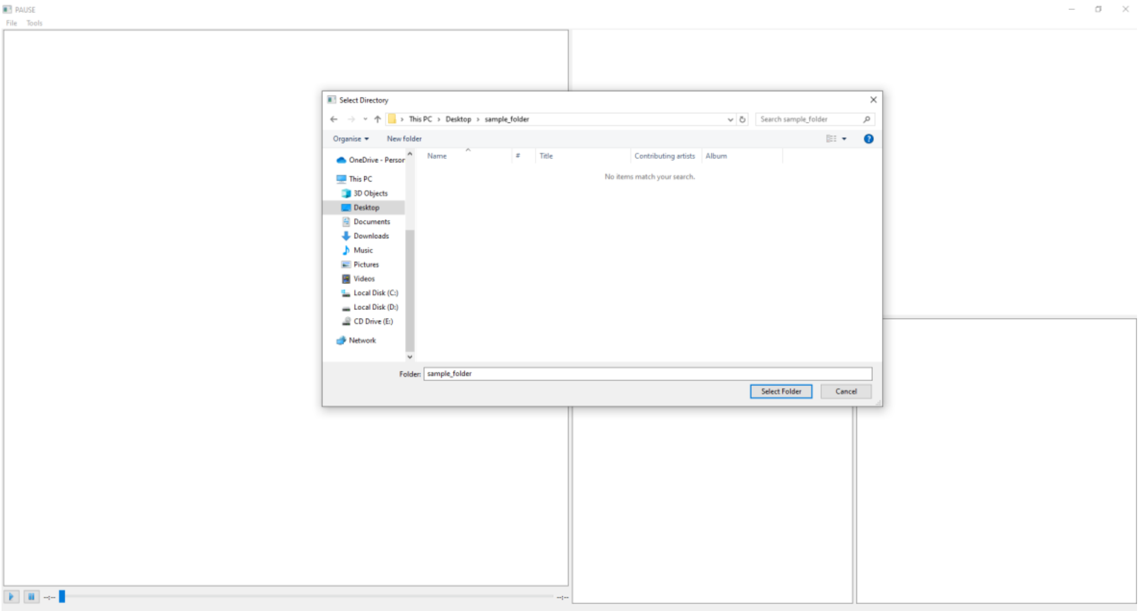
Analysis has finished

Visualize the results by selecting an option below

Figure 29 Analysis finalised prompt

Finally, let us have a look at the remaining options under the “File” menu. These options all relate to saving results or loading previous results to avoid the necessity of redoing the analysis. In all cases, a browsing window will be opened to select the files to be loaded or to select the folder in which the user wishes to save the results. The first option is called “Save annotations”. This option will save the annotated text in a .txt file inside the selected folder. The second option, “Save graphs” will do the same for the different plots that have been generated during the work session. These plots, in order for them to still be interactive, will be saved as .html instead of as images. These two options will save the results in human understandable formats. However, option “Save project” will serialize the project and save in at as a JSON object. This format is not human-readable, but weight less and thus is a handy tool to save large projects in a form that does not occupy a lot of disk space and that can be easily converted into its original form by the program.

The last option is “Load project”. This option will take a project previously saved as a JSON object and display the previous work so that the user can continue where they left.



No items match your search.

Folder: sample_folder

Figure 30 Save/load window

4. Use cases

In this section we will describe and explain the different interactions that the user can have with PAUSE. We will do this going through the use of each of the three annotation processes. In this sense, this section may also be understood as a basic guide to the use of the program.

The end users for which the tool has been designed are researchers in the fields of linguistics, literary or philological studies. For this reason, in the use cases here described "user" and "researcher" are used interchangeably. Each use case is centred around the request by the user to the program of the annotation of a given text. As seen in fig. 31, the use cases that we consider are the use of each of the three annotation systems. There are also explanations detailing how these different cases may interact with each other.

All three use cases share some common basic aspects. In all of them, the main actor is the researcher. As a user, the researcher can choose to provide a poem in both written and voice-recorded formats. Then the *Caesuras and enjambments, Paralinguistic features* or *Optophonetic notation* may be selected to generate the desired output using the written form of the poem as a layout.

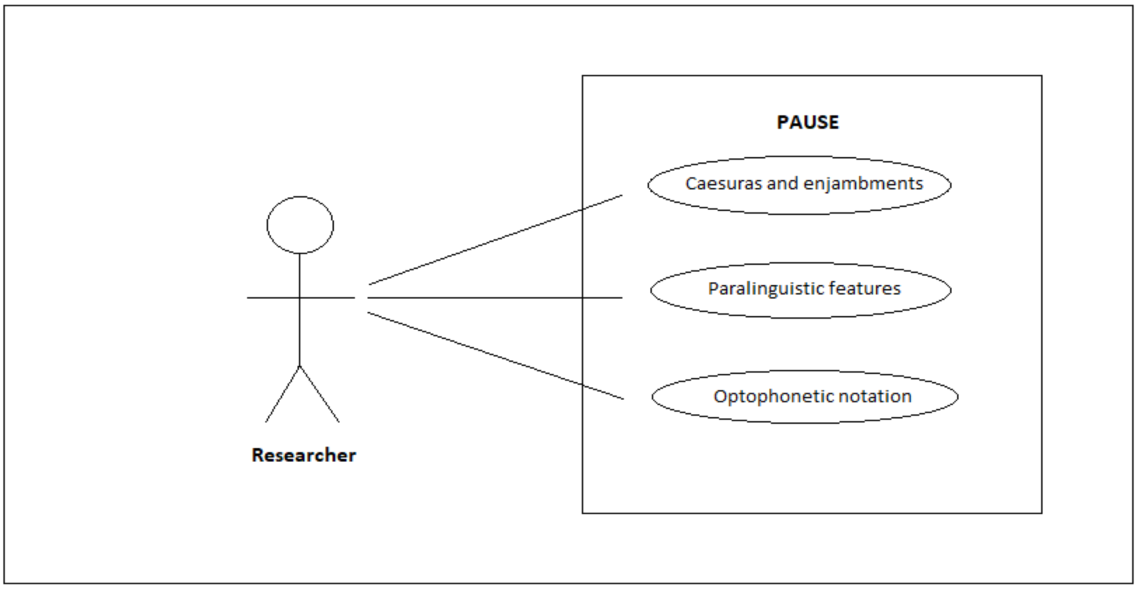


Figure 31 Use case for PAUSE

4.1. Caesuras and enjambments

In his scenario the researcher’s purpose is to annotate the caesuras and enjambments of a poem as they are uttered in the provided recording.

USE CASE: Annotating the caesuras and enjambments.

PRIMARY ACTOR: The researcher.

INTEREST: To obtain an annotated text displaying the use and absence of pauses in a specific reading of a poem.

NEEDS: The annotation must be precise and unambiguous.

The annotated text must be interactable to allow for modifications and corrections.

The parameters must be modifiable to allow decision making and fine tuning.

INTERACTIONS	
STEP	**ACTION**
1	Before starting the process, the user must have a textual transcription of the poem, as well as an audio recording of the same poem.
2	The user opens PAUSE and selects “File/Open text file” and selects the text of the poem. Then selects “File/Open audio file” and selects the recording. These two actions are interchangeable in order.
3	The user selects “tools/annotations/caesuras & enjambments”.

4	PAUSE analyses the recording, calculates the parameters to determine what constitutes a pause or an absence of a pause and annotates the poem accordingly.
5	The user interacts with the output of the previous step to apply any modifications.
6	The user decides whether or not they agree with the value of the parameters used for the annotation. They may now decide to change them and repeat the annotation.
7	The user saves the result of the annotation.

Table 3 steps of Caesuras & Enjambments

EXAMPLE

Here follows a practical exemplification of the steps explained above. Let us imagine a researcher called Gianluca who wants to use PAUSE for the annotation of a poem. He has a recording an out loud reading of the poem *Alto Jornal,* by Spanish poet Claudio Rodríguez, and he wants to annotate the written poem to graphically mark the caesuras and enjambments as they are uttered in the recording.

In this scenario Gianluca fulfils the requisites mentioned in step 1. That is, he has a written poem and a recording of said poem. Now, he must load those files into the program. In fig. 32 he is selecting “File” in the menu, and then, in fig. 33 he has clicked “Open text file” and he is navigating the file explorer to find the poem, in .txt format and opening it.

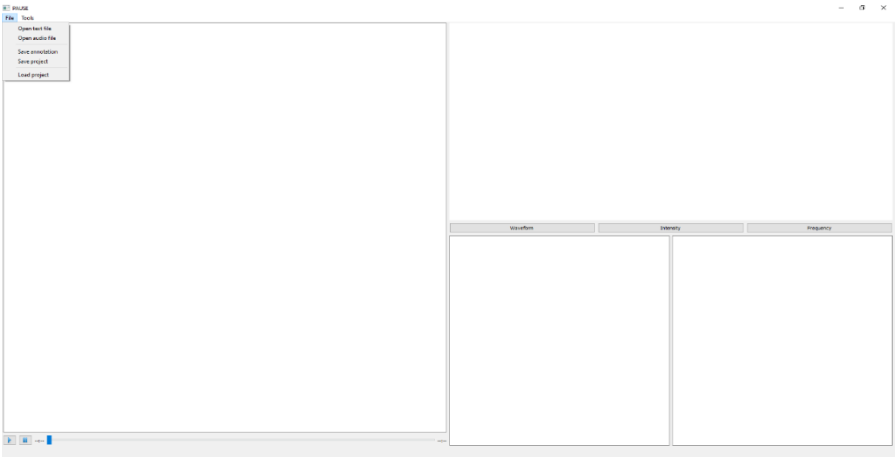
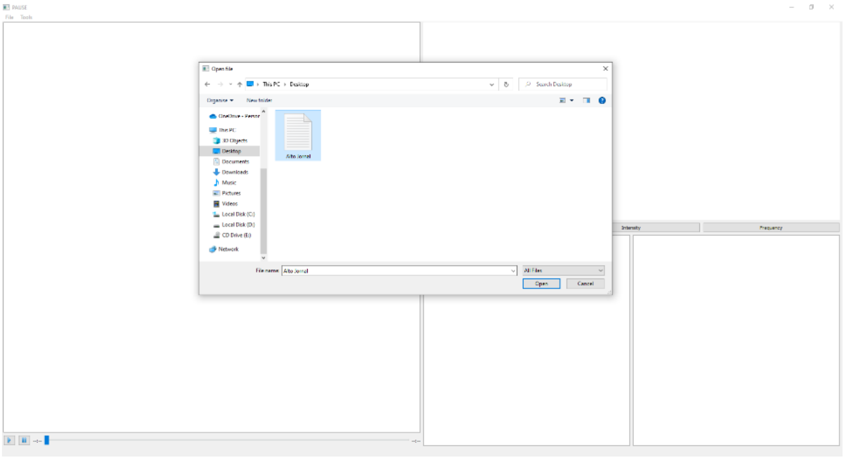


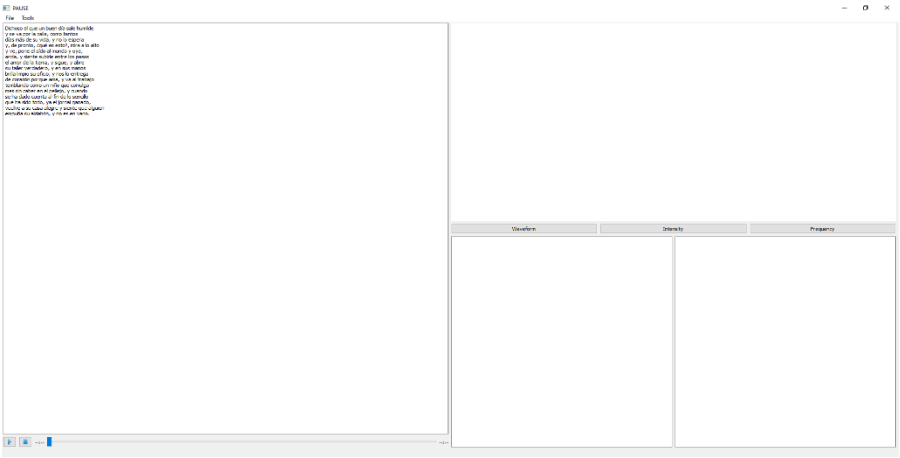
Figure 32 File.



File name: Alto Jornal

Figure 33 Open text file.

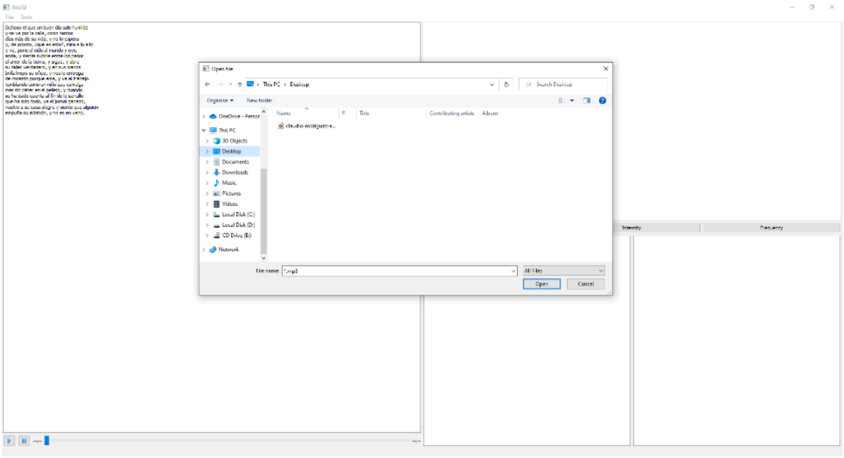
Now that he has opened the file, the text will appear on screen as shown in fig. 34.



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 34 Display of poem.

Now that he has opened the text file, is time to repeat the step represented in fig. 32, but this time selection “Open audio file”. In fig. 35 we can see he is navigating the file explorer to open the audio file, this time in .mp3 format.

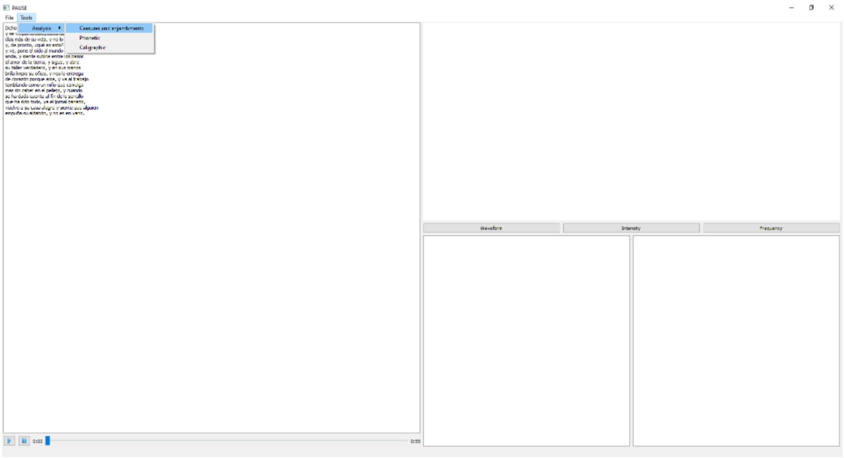


Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

File name: *.mp3

Figure 35 Open audio file.

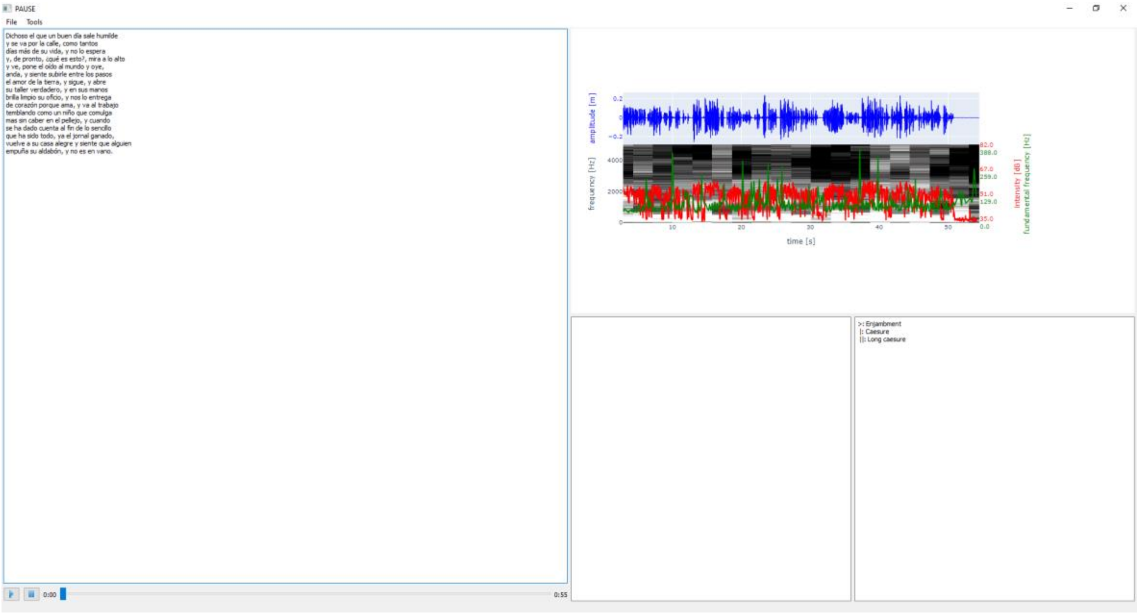
Next, is time to select the desired annotation system. In fig. 36, Gianluca is selecting “Tools/Analysis/Caesuras and enjambments” for annotating the poem. As a side note, in fig. 36 we can also see in the bottom left part of the image that the duration of the audio has appeared after opening the audio file. If he now wishes to, Gianluca may press the play button to star playing the audio.



Dicho
días más de su vida, y no lo
y, de pronto, ¿qué es esto?
y ve, pone el oído al mundo
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 36 Selecting notation.

Pressing that button will offer him a view of a combined graph of the waveform, pitch and intensity of the audio recording in the style of PRAAT:



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

>: Enjambment
|: Caesure
||: Long caesure

Figure 37 Waveform, pitch & intensity graph

4.2. Paralinguistic features

This scenario is perhaps the more linguistic or empirical oriented approach to the study of the poem of interest. The researcher in this case seeks to obtain a formal notation of the pitch, speed, and volume of the poem’s audio recording. Useful graphs are also provided to expand and contextualize the data used for the annotation process.

USE CASE: Annotating the paralinguistic features of speech.

PRIMARY ACTOR: The researcher.

INTERESTS: To obtain an annotated text displaying some important paralinguistic features of speech (volume, pitch, and speed), as well as graph complementing the annotation.

NEEDS: The annotation must be precise and unambiguous.

The annotated text must be interactable to allow for modifications and corrections.

The parameters must be modifiable to allow decision making and fine tuning.

The graphs must be interactive to allow for contextualization (e.g., scrollable).

INTERACTIONS	
STEPS	**ACTIONS**
1	Before starting the process, the user must have a textual transcription of the poem, as well as an audio recording of the same poem.
2	The user opens PAUSE and selects “files/add text” and selects the text of the poem. Then selects “files/add audio” and selects the recording. These two actions are interchangeable in order.
3	The user selects “tools/annotations/paralinguistic features”.
4	PAUSE analyses the recording, calculates the parameters to determine, for each of them, the appropriate statistics and normalization. Then the poem is annotated using the metrics obtained in the analysis.
5	The values of the obtained metrics, as well as their normalization and relevant statistics, are plotted to give the researcher the information needed to properly read and understand the output of the analysis.
6	The user interacts with the output of the two previous steps to apply any modifications.

7	The user decides whether or not they agree with the value of the parameters used for the annotation. They may now decide to change them and repeat the annotation.
8	The user saves the result of the annotation.

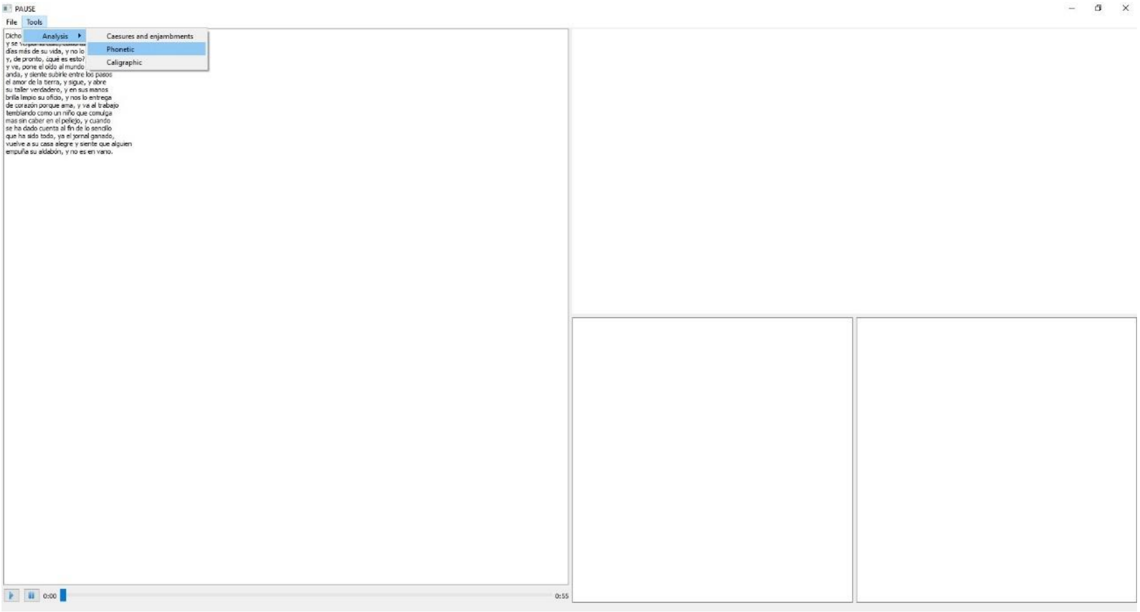
Table 4 steps of Paralinguistic Features

EXAMPLE

As in the previous section, we will now provide a practical exemplification of the use of this notation. Let us imagine a researcher called Gianluca who wants to use PAUSE for the annotation of a poem. He has a recording an out loud reading of the poem *Alto Jornal,* by Spanish poet Claudio Rodríguez, and he wants to annotate the written poem to annotate into the text notions of pitch, intensity and speed as per the analysis carried out by the program.

Now Gianluca fulfils again the requisites mentioned in step 1. That is, he has a written poem and a recording of said poem. Now, he must load those files into the program. In fig. 32 he is selecting "File" in the menu, and then, in fig. 33 he has clicked "Open text file" and he is navigating the file explorer to find the poem, in .txt format and opening it. The steps he follows now are as described in the previous section. He will also load the audio file of the recording in the same way as shown before.

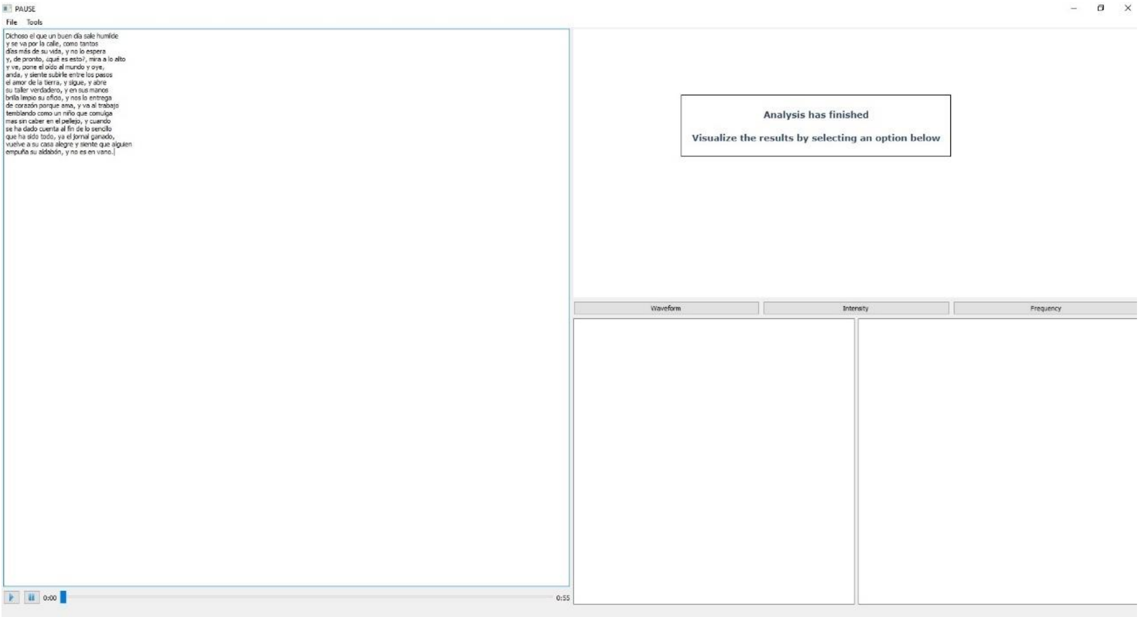
Now, instead of selecting "Tools/Analysis/Caesuras and enjambments", he will press the button "Tools/Analysis/Phonetic" as shown in fig. 38:



Dicho
y se
días más de su vida, y no lo
y, de pronto, ¿qué es esto?
y ve, pone el oído al mundo
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 38 Selecting notation: Phonetic

Once the tool's analysis has finalized, he will be prompted by a notification stating so as seen in fig 39:



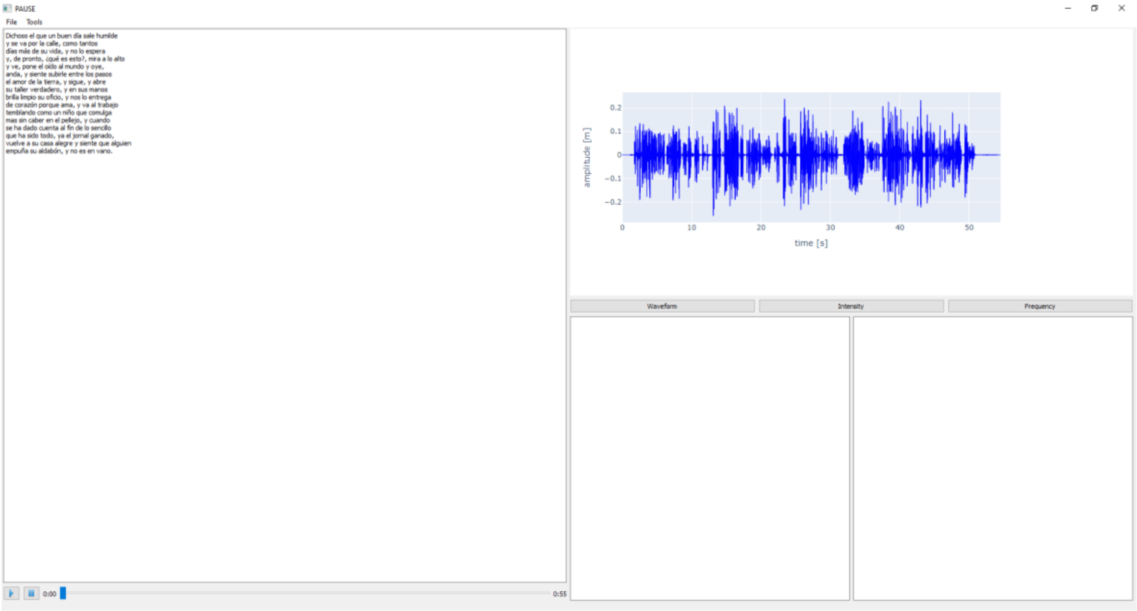
Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Analysis has finished

Visualize the results by selecting an option below

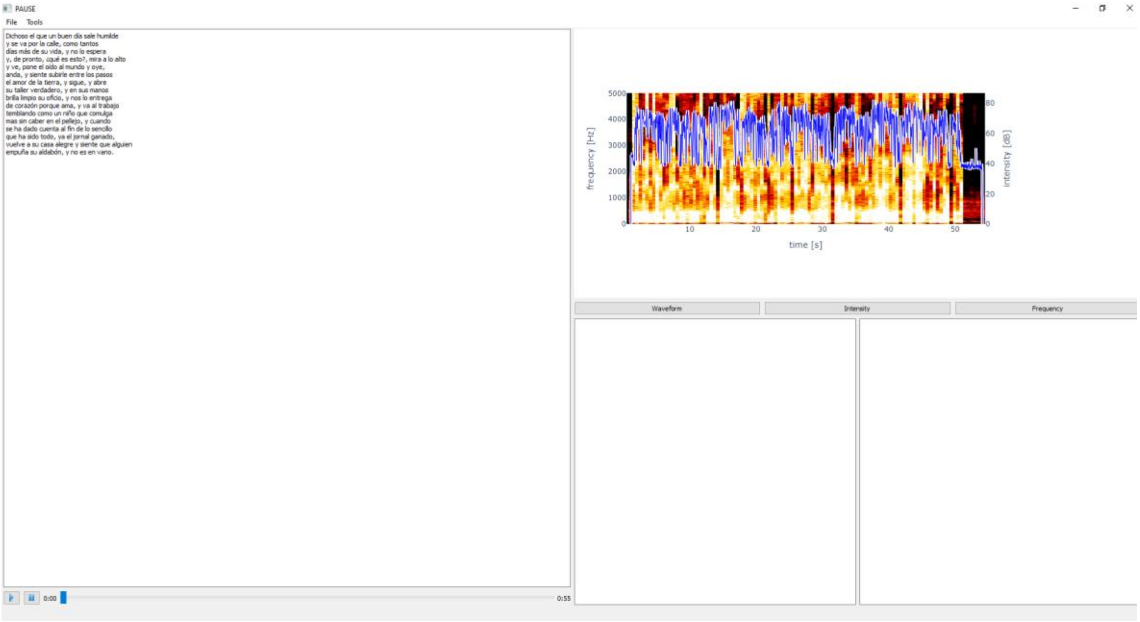
Figure 39 Analysis completed

As instructed by the prompt, selecting one of the three possible graphs will display them, and he will be able to switch between them on demand.



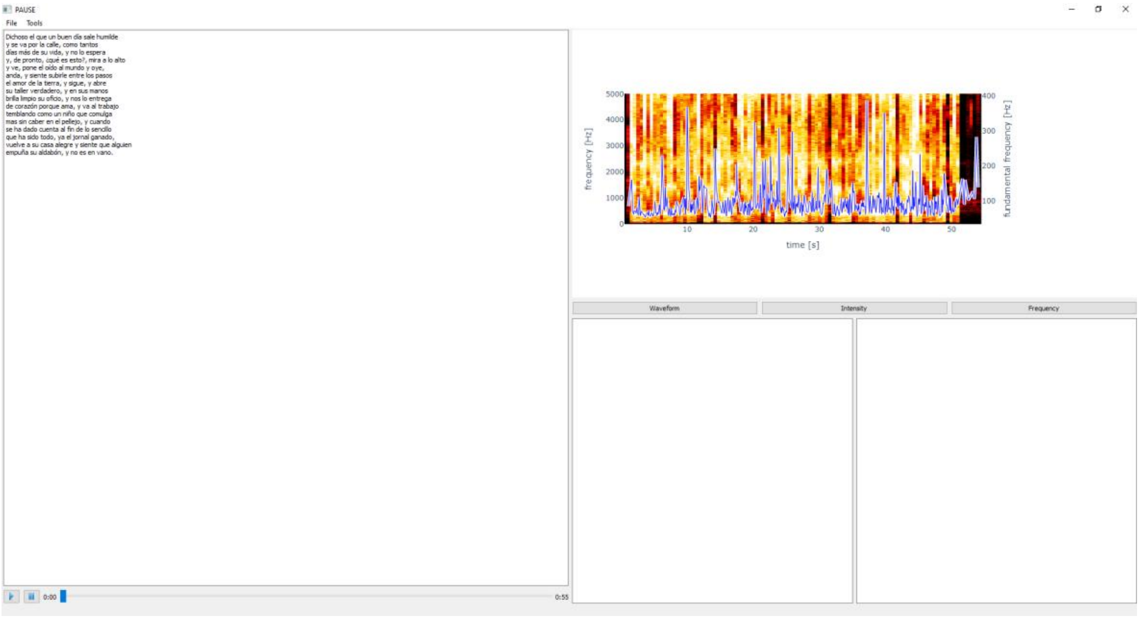
Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 40 Screen with waveform plot



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 41 Screen with intensity plot



Dichoso el que un buen día sale humilde
y se va por la calle, como tantos
días más de su vida, y no lo espera
y, de pronto, ¿qué es esto?, mira a lo alto
y ve, pone el oído al mundo y oye,
anda, y siente subirle entre los pasos
el amor de la tierra, y sigue, y abre
su taller verdadero, y en sus manos
brilla limpio su oficio, y nos lo entrega
de corazón porque ama, y va al trabajo
temblando como un niño que comulga
mas sin caber en el pellejo, y cuando
se ha dado cuenta al fin de lo sencillo
que ha sido todo, ya el jornal ganado,
vuelve a su casa alegre y siente que alguien
empuña su aldabón, y no es en vano.

Figure 42 Screen with frequency plot

4.3. Optophonetic annotation

In this last scenario, the researcher wants to get an informal, yet still rich, annotation of the given poem. This case is more oriented towards teaching or dissemination. The goal is to obtain an annotated poem that displays the relevant information through visual elements, like the colour, size or font of the words.

USE CASE: Creating an optophonetic version of the poem.

PRIMARY ACTOR: The researcher.

INTEREST: To create a sort of optophonetic version of the poem that displays the same information the two others notation show, but in a less precise way. The information here is displayed through the colour, shape, font and size of words, as well as through the verse structure and separation between words.

NEEDS: The annotation must be visually pleasant without losing richness.

The annotated text must be interactable to allow for modifications and corrections.

The parameters must be modifiable to allow decision making and fine tuning.

INTERACTIONS	
STEPS	**ACTIONS**
1	Before starting the process, the user must have a textual transcription of the poem, as well as an audio recording of the same poem.
2	The user opens PAUSE and selects “files/add text” and selects the text of the poem. Then selects “files/add audio” and selects the recording. These two actions are interchangeable in order.
3	The user selects “tools/annotations/optophonetic annotation”.
4	PAUSE analyses the recording, calculates the parameters to determine, for each of them, the appropriate statistics and normalization. Then the poem is annotated using the metrics obtained in the analysis.
5	The user interacts with the output of the previous step to apply any modifications.
6	The user decides whether or not they agree with the value of the parameters used for the annotation. They may now decide to change them and repeat the annotation.
7	The user saves the result of the annotation.

Table 5 steps of Optophonetic Annotation

EXAMPLE

We will not provide an example for this notation, because the steps to follow and display of the plots is very similar to the phonetic one. To have an exemplified guide on the use and visualization, please refer to the previous subsection.

4.4. Remarks and considerations

The previous sections offer an overview of the use of PAUSE for a single annotation. That is, they focus on scenarios in which the researcher wants to annotate a single text, just once.

Although this may be the case in some situations, it is in the researcher's interest to have the ability to perform a comparative work of many texts or annotations of the same text. The most obvious comparative work that arises from the previously detailed use cases is working with two or more kinds of annotations (e.g., caesuras and enjambments, and paralinguistic features). But it is interesting to work with many different notations of the same text and kind but with different values for some, or all, of the parameters. In this case, for instance, modifying some parameters but fixing others across annotations allows the researcher to create control variables for the study of the influence of said parameters. Lastly, it could also be interesting to study annotations of different versions or witnesses of the same poem.

To achieve this, the user can save every annotation that they consider to be of interest. Then, all the saved annotations can be shown alongside the current working session to allow for comparisons.

5. Analysis

In this section we will discuss how the analyses that are shown in the tool are performed. The input for the analyses is a pair conformed by a written poem and the corresponding recording of its recital. From the audio we perform the necessary plots, and then an annotation of the text is performed.

This section will be slightly more technical on the computational side, so users might still benefit from the tool without understanding every detail of it. However, to get an in-depth knowledge of the results obtained, it is useful to know the nuances of how they are found. Consequently, readers can use the definitions provided on the first section as an aid to the understanding of this section.

We will first explore how the data is transformed and analysed to obtain all the elements that are required to later create the plots and annotations. Then, we will deep dive on which plots are created and how they can be interpreted. Finally, we will detail how the annotations should be created.

5.1. Data cleaning and analysis

It is very common that, when analysing information, the provided data needs to be transformed and adapted to make it cleaner and more relevant. This process is known as data cleaning. Later, relevant features can be extracted from the cleaned data, when we perform data analysis.

Because the input is raw data, we need to first perform these tasks. First, the text and audio files are modified to meet the requirements of the libraries that are later used in a cleaning step. Then, we move to the analysis step. We extract information from the cleaned audio and text file, such as voice intensity or amplitude, to later create the plots and annotations.

Let us first look at the cleaning steps. As it has been mentioned in previous sections, all analyses in the tool are performed using as basis a text of a poem and an audio of such poem recited. The text needs to be a .txt file where each line contains a single verse. If the audio does not recite the full poem, the text should only include the recited part. In line with this requirement, the title should be excluded. This is to ensure a full match between the text and the audio.

We only clean the audio file, as the text file is assumed to be already consistent and clean. The audio is loaded into the tool as an .mp3 file. This is the most common audio format, so it should be the most accessible for users. However, the libraries we will be using require audio to be in .wav format, so we first transform it by using the pydub python library (Robert et al., 2018).

Audio files are usually noisy. This means that they are 'polluted' with background noise, interferences from audio recorders, etc. This noise makes it difficult to extract the relevant features to perform our analyses. In general, all the noise we want to reduce is categorized as background noise. Background noise (Panayiotou and Bon, 2023) is any undesired auditory stimuli that co-occurs with the voice we are trying to analyse and is usually caused by ambient noise. It modifies the speech waveform and can impact its components, such as frequency or amplitude. As we will be studying those in our tool, misleading values in them can lead to faulty results. Finding the best filtering technique is something that is done iteratively, through try and error. Because we have not fully implemented the analysis, we do not have the necessary output to compare the impact of different methodologies and determine which works best for us.

There are multiple methods that can be used to reduce background noise, ranging from filters, such as adaptive filtering algorithms and Wiener filter; to complex machine learning techniques, such as deep learning models and GANs. One simple yet powerful technique is to apply a high-pass and a low-pass filter. These filters remove audio below and above a certain frequency. High frequencies can be caused by electrical noise or hiss. Lower ones can be caused by hum or rumble. By removing them, we keep a spectrum of frequencies in the middle range, where the human voice usually lays.

For some use cases of speech recognition, it is desirable to normalize some aspects of the data. This is done to try to make voices sound according to a standard to simplify some tasks. For example, to be able to understand different accents. However, these particularities are crucial to us. Removing them would mean to remove the object of our study almost altogether. Because of that, we must not select a technique that modifies them.

After these steps we can consider that data cleaning is done, and that we have our audio file available in the format that is required and without any undesired noise in it. It is time to extract the features that will lately be used to create the plots and annotations. Here we describe how the wavelength, intensity, fundamental frequency and spectrogram values are extracted from the audio file. To further enhance the tool in the future, more features might need to be extracted to create additional plots or better automate the annotation process based on experts' knowledge and user feedback.

To perform these feature extractions, we will use the python library Parselmouth. Parselmouth is a wrapper of internal Praat code so that it can be used in python. It ports the functionalities of Praat into python so that in can be used in our tool with the same outputs as if we were using Praat directly. Our choice for the library to extract features is determined mainly by this, as Praat is the go-to tool in the research community when it is required to perform phonetic analyses.

Even if Parselmouth currently does not support every Praat functionality, there are enough present to fit our purpose. Also, the author mentions that the project is still in active development. Any further release of Parselmouth should be closely monitored to check if new features can aid with expanding and improving our current tool.

Let us start with the wavelength extraction from the audio file. This can be done directly. However, the result is large and noisy, which makes it hard to, when plotted, extract any useful information from it. If every small variation of the wavelength is captured, it becomes hard to see global tendencies or patterns in the result.

To avoid this, we resample the audio file before extracting the wavelength. Instead of obtaining the wavelength at every possible moment of the audio file, we average very short segments of the audio before doing so. This way, the resulting wavelength will only contain these averages, reducing the number of unnecessary spikes and allowing us to see global patterns. Keep in mind that, even if we are averaging the properties of the audio file, the duration of the segments where the average is done is very small. The final result will still have a sense of continuity but will be much easier to analyse.

Intensity is very direct to extract, similarly to wavelength, as Parselmouth offers functionalities to directly extract it from the audio file in .wav format.

Frequency can also be directly extracted, and fundamental frequency from the frequency too. However, the result is not satisfactory, as it captures noise that is hard to remove and fails to show clear patterns on the data. Because of that, we apply interpolating and smoothing techniques to the results. Interpolation (Steffensen, 1950) allows to fill the gaps between the data that we know with the behaviour that it is believed to have. If some frequency has not been able to be obtained, we will check closer frequency values in time and deduct the one that is missing. Using this, we reduce the number of missing data that we encounter finding the fundamental frequency from the frequency directly. Then, smoothing (Simonoff, 1996) removes the noise that was captured when the fundamental frequency was found directly. This is done by making the points that are very different from the adjacent ones closer to them, as they are probably caused by noise. When noise is hidden, important patterns in data that were previously disguised by sudden fluctuations arise.

Finally, spectrogram values are found. They behave similarly to wavelength. They can be extracted directly, but the result is noisy as well, with very sudden variations that difficult finding global patterns. We wish to average them, just as we did with the wavelength. However, spectrogram values have two dimensions, frequency and time. Because of that, the average needs to be performed in both dimensions to ensure clear patterns can be found.

Once these features are extracted, we are ready to plot them and use them to create annotations. As we mentioned before, if we wish to expand the tool’s functionalities, more features could be added to this list. It is also important to find better techniques for all the steps mentioned in order to improve the quality of the results.

5.2. Plots

The easier way to find patterns in the data that has been extracted in the previous section is to visualize it through plots. Python offers several libraries to plot data. The most common ones are matplotlib (Hunter, 2007), seaborn (Waskom, 2021) and plotly (Plotly Technologies Inc., 2015).

We have decided to use plotly for the tool because of the properties of the plots that it can produce. Matplotlib and seaborn can generate static images, the ones we are used to find when we see graphs or other similar representations. However, plotly allows us to interact with the plot once it has been generated. We can zoom in an out, hover over it to see data's numerical value, scroll over a zoomed plot to view different sections of it, etc. This is possible because matplotlib and seaborn produce images that can later be saved in jpeg, png, or other image formats. However, plotly produces HTML objects that can be interacted with.

This interactivity is key for the user experience. For example, when a user wants to analyse a particularly long poem, they can zoom in specific sections to see how wavelength, intensity, etc. vary. They can also check a specific value for one property just hovering over it. This allows faster and more accurate analyses from plots.

Also, if in the future it is desired that the tool becomes a web app, dash (Hossain, 2019) can be used. Dash is a python framework to build applications that display data and plots and is built on top of plotly. This means that dash is specifically created so that plotly visualizations can be used in web development easily, ensuring maximum compatibility between our plots and the web app.

Now let us examine the plots that are present in the tool. There are four of them: wavelength, intensity, frequency and a combined plot of the previous three ones.

We start by taking a look at the wavelength plot. We just display the wavelength value at each timestamp. The further the amplitude at a certain timestamp is from zero, the greater the amplitude of the audio at that timestamp is. For example, during silences, we expect the amplitude to be very close to zero.

Ideally, its value would be exactly zero, but we need to account for some remaining noise in the audio file. We can check its value for a certain timestamp by hovering over the blue line at the desired timestamp.

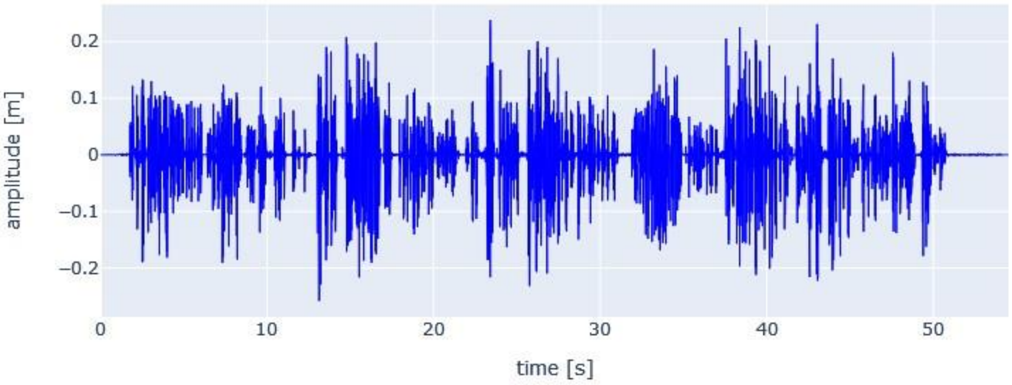


Figure 43 Waveform plot

This plot offers a view into the general behaviour of the audio signal. It helps with having an idea of the intensity patterns, as well as were the silences fall. It has to axes with time (s) and amplitude (m).

Next, the intensity plot displays the intensity of the audio using a blue line. As background of the plot, we display the spectrogram values using a scale that ranges from very light yellow to very dark red. Intensity values at a certain timestamp can be retrieved by hovering over the blue line. However, spectrogram exact values are not available. We have decided to not show them to avoid excessive clutter in the plot, as we believe that it is more useful to display the pattern that it follows rather than their specific values. It has three axes, displaying time (s), frequency (Hz), and intensity (dB).

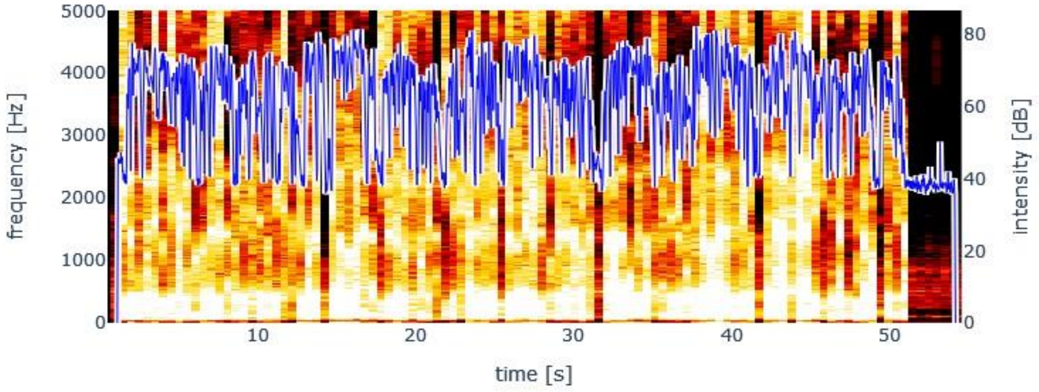


Figure 44 Intensity plot

We use this plot in both the phonetic and optophonetic analysis as a visual support for them. This plots the values used for the notations. The fact that the user can hoover over it to see specific values offers more insight into the analysis themselves.

The frequency plot is very similar to the intensity plot, where instead of showing the intensity, the fundamental frequency is displayed.

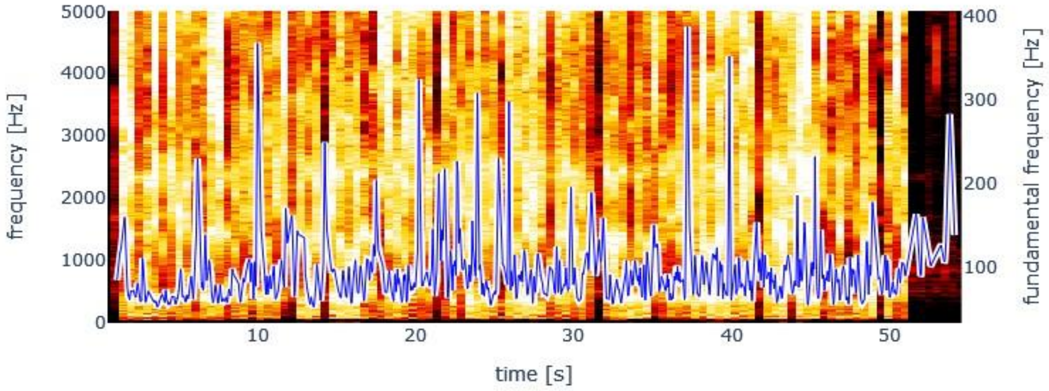


Figure 45 Frequency plot

We use this plot to visually display the data used for the pitch and optophonetic annotations, alongside the result of the notations.

The final plot is a combination of the previous three. It is divided horizontally into two subplots. The top one is almost identical to the wavelength plot. The bottom one combines the intensity and frequency plots into a single one. Intensity and fundamental frequency are represented in red and green respectively. As they are plotted in the same axis, their scale matches their colour to ease interpretation. The spectrogram is also available on the bottom plot. It is plotted in grayscale to avoid using an excessive number of colours that might make it difficult to interpret the plot.

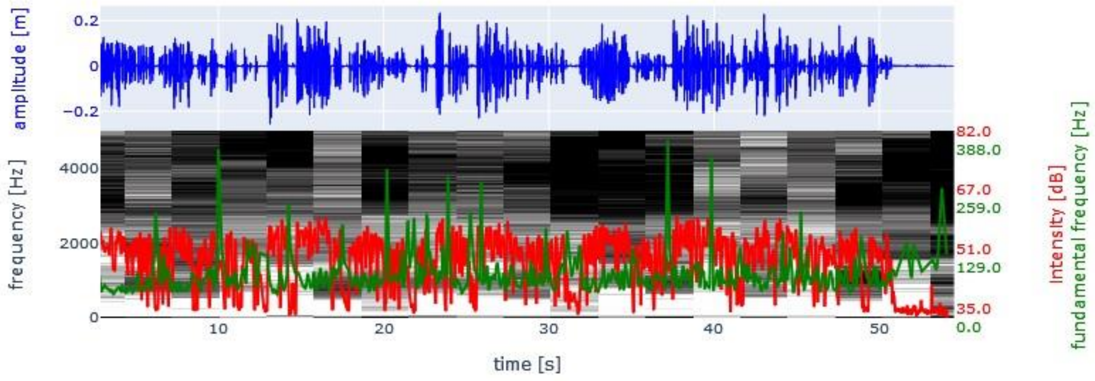


Figure 46 Combined plot

It is very useful to be able to compare the behaviour of amplitude, intensity and fundamental frequency together to extract patterns in the audio for any analysis.

5.3. Annotations

Let us discuss one by one the different notations. As discussed, the caesuras and enjambments notation aims to determine the poem structure that has been given by the reader’s voice compared to the one that is laid out in the poem text.

Consequently, we want to find pauses in the audio file to be able to compare them with the ones we would expect based on the poem's textual markers. We believe that the optimal way of finding a method to automatically annotate a text is to perform such annotations manually and iteratively, until an accurate, methodical and replicable process is found. We use *Alto Jornal*, by Claudio Rodríguez to test possible methodologies manually that could later be automated in our tool.

As mentioned, we desire to find pauses when the author recites the analysed poem. To determine such pauses, we first apply a frequency and intensity filter to our audio file. This way we remove undesired values that would otherwise skew the analysis. After, we can classify audio segments based on whether they are silences or not.

Once the silences are available, we need to determine whether they are a long or short pause for each of them. To do so, a threshold needs to be defined. Every pause longer that the threshold will be considered a long pause and the ones equal or shorter than the threshold will be considered short pauses. There is a wide range of methodologies that could be used to find the threshold value, but they can be classified into two groups: statistical and perceptive. Statistical methods define the threshold as a statistical measure that captures what is the average or the typical silence duration. Examples would include the mean, median or a certain percentile of the silence duration. On the other hand, perceptive methods are subjective. The user needs to determine the threshold considering the purpose of their analysis.

We believe that perceptive methods are more suitable for this use case. The main advantage of statistical methods is that they are automatic and produce a result without the user's intervention. However, such result might contradict the natural perception of the listener and are very contextual. For example, if the author tends to do very long pauses when reading their poems, a threshold found by the statistical method may determine that pauses that are clearly long to us are, in fact, short according due to the existence of few, shorter pauses that correspond to phenomena of natural speech. Contrarily, perceptive methods do not have this inconvenience, as the threshold is set manually. Their main drawback is that the threshold might need to be changed by the user. However, a manual input for the threshold should be easy to implement in our tool.

The threshold for our analysis of poem *Alto Jornal* has been set to 0.2 seconds. This value is consistent with what is proposed in Mistrorigo (2018), where the caesuras and enjambments notation is introduced. This supports our hypothesis that a manual selection of the threshold is a good option to divide between long and short pauses.

The next step is to determine whether the pauses that we have found in the audio file are expected. To do so we have to parse the tex. Then, we check if a textual mark of pause exists in the text where a pause is found in the audio. Textual marks are punctuation symbols and end of verses. If a pause coincides with one of those, then it will be marked as an expected pause, otherwise, as unexpected. Finally, we analyse whether the pauses that we would expect based on the text's textual marks are in fact present in the audio. In case we find a textual mark for a pause that does not correspond to the boundary of a silence, it has to be marked as an enjambment.

Now let us move on to three analyses needed for the phonetic notation. First, we have the pitch analysis. The first step to perform this annotation is to be able to match text and audio using forced alignment. As we will discuss below, we have been unable to perform this task automatically. Consequently, we had to manually label tones. However, once forced alignment task is successful, it is possible to perform this annotation automatically thanks to several scripts that are available for this or very similar tasks (Elvira-Garcia et al. 2016). However, a close analysis of the proposed script revealed unexpected and potentially undesired behaviours of the code. The algorithm itself should also be adapted to fit our exact porpoise. Hence, a proper adaptation and testing of the resulting script is needed for the tool.

These algorithms work by finding the range of the voice in the recording and discretizing the values in that range. Then, different positions of the phrases are associated to the respective tier. Because we are only interested on tiers 3 and 4 of Sp_ToBI, that would ease the classification. Then, according to the tier and value of the pitch, a tag is assigned.

Then we have the intensity analysis. If we check the intensity representations for the human voice, we will soon realise that there are constant intensity variations. The intensity contour is not a monotonous line, but rather “spiky” instead. The first part of the analysis, then, will be to smooth the values. We have already mentioned different ways of doing that. Although, due to the complexity of the typical intensity function, some further steps may be needed. We recommend using Kernel Density Estimation (KDE) (Conlen, n.d.) for this task. KDE is a complex algorithm that weights the values of neighbouring data to create a curve with less ups and downs. It is an evolution of the methods that we have already applied. This allows us to get information of the tendency of the contour. This way, we clean out non relevant local maximums and minimums to study these tendencies.

Once we have the smooth data, we simply have to assign the symbols explained in section 2 to the intervals that follow an upwards or downwards tendency.

Now, to the last of the set of three analyses for the phonetic notation, the tempo analysis. To determine which fragments of the poem are fast and slow and, consequently, be able to annotate them as such, we need to define a measurement of speed. We have decided to move forward with a mathematical formulation of speed, as number of phonemes in a fragment divided by the duration of such fragment in seconds. Even if it might not capture all phenomena that we associated with speed, as discussed before, we believe it will provide insights to the users of our tool.

The calculation of the speed of each fragment is very straightforward once we have them delimited. Those are available from our previous annotations on caesuras and enjambments, including their durations in seconds. We only need to count the number of phonemes in each of them to find their velocity. There are numerous tools to determine the distinct phonemes in a speech recording, including some tools that also perform forced alignment and will be discussed in the next section.

The next step is to find which tempos are fast, average and slow. To do so, we define what we consider to be average speed. Every speed above it will be annotated as fast, and all below it will be annotated as slow. We first find the average and the standard deviation of the speeds of all fragments. We then define average speed as any speed between the average minus one standard deviation and the average plus one standard deviation. Everything that falls in the interval defined as (average ± SD) will be considered average. Then, we draw the corresponding line over the fragments that fall out of this interval.

Lastly, let us talk about the optophonetic notation. This one reuses the calculation and analysis detailed in the previous paragraphs. In fact, it can be seen as a simplification of the phonetic notation, in conjunction with an adaptation of the caesuras and enjambments notation. The most notable change is the simplification of the pitch annotation system. Due to the final projected form of this annotation system, we have decided o only represent H and L tones. For this reason, M tones can be ignored altogether.

5.4. Remarks on forced alignment

We have tried several packages to perform the forced alignment task. Here we wish to provide the reasons why they failed, to avoid future duplicated work during future development of the tool.

The first option was Praatalign (Lubbers and Torreira, 2013-2018). We believed it to be the best approach in theory as it is a library that allows to use the forced alignment functionalities of Praat in python, similarly to what Parselmouth, the library we have used in data analysis, does. However, Praatalign only works on python 2 and our tool is developed in python 3. We believe our tool should not use python 2, as it is no longer supported (Python Software Foundation, 2020). This means that if an error is found in python 2, it will no longer be fixed. Also, improvements that come into python 3 in the future will not be included in python 2. To ensure the correct maintainability of our tool, it should not use python 2.

Next, we explored Sail Align (Katsamanis, Georgiou, Goldstein and Arayanan, 2011). We quickly found that it is built to only work on Linux systems. Even if Linux is the most common system for developers, users usually work on

Windows. To ensure that our tool can be used by the audience that would benefit the most from it, we decided to discard this library. If the tool is moved to a webpage in the future to ease its access, the use of this library could be reconsidered.

After that, we tested the first library that was compatible with our python version and operating system: Aenas (Read Beyond, 2017). This library takes a text file and an audio file and produces a synchronization map between text fragments in the text file and the timestamps in the audio file when the text fragment is recited. We tested the library using poem *Alto Jornal* by Claudio Rodríguez and the results were not satisfactory. When we manually checked the map, we found that the text fragments were not recited in the audio when the map suggested. Consequently, we had to discard this library, as it does not produce accurate results.

Then, we explored library PocketSphinx (Carnegie Mellon University, 2023). It promised that it could perform the forced alignment task we required in its documentation. When we tested it, we found that the library allows users to find the places where it is most likely to find a word in an audio. This does not suit our purpose, as one word might be repeated multiple times in a poem, and we need to map verses as a whole to analyse their structure.

Finally, we checked SPPAS (Bigi, 2016). This library seemed very promising, as it is used by research institutions and is updated regularly. We followed their detailed instructions to perform forced alignment using both python and the command line interface. In both cases we found the same cryptic error that did not allow to continue with the alignment process.

For these reasons, we were not able to conclude the implementation of forced alignment. Because this is a crucial step that is needed to implement the rest of our proposed algorithms, this means that we could not test them. Their implementation and testing must be then considered as future work.

6. Conclusions and future work

This thesis discusses a proposal for a computational tool that aims to be of use for the research community, both in linguistics and literary criticism fields. It is a digital humanities project that makes use of state-of-the-art software and computational techniques to analyse audio and text data to annotate the input text. It is based on the theoretical framework proposed by authors like Newell-Smith or Mistrorigo, that consider poetry as a multimodal art form in which the voice of the reciter plays a structural role in the shaping of the poem.

We began by looking into the existing bibliography on the various fields and topics that are relevant to this project. Due to its multimodality, we explored topics related to computer science, computational linguistics, phonetics, digital humanities and literary criticism. We paid special attention to some terms and topics that are important to understand every detail of the project and that may not be fully well-known to some readers. Due to, again, its multimodality, we want to ensure that we provide the means to understand the whole project to readers coming from different backgrounds and specializations. We also discussed the potential benefits of making use of artificial intelligence, as well as pointed out its drawbacks. We discussed why, in its current form, it is not a beneficial implementation to the tool. However, it may be in the future. We deemed this discussion important because, as researchers, it is important to take hot topics into account and argue against their misuse.

We then moved on to the proposals of the various annotations. We have presented three annotations. The first one is based on the works in Mistrorigo (2018) and annotates pauses and the lack of pauses in certain points of the recitals of poems. This refers mainly to two phenomena known as caesuras and enjambments, which play an important role in the performance and interpretation of poems. The second annotation is for three features of prosody: pitch, intensity and speed. We divided the discussion of this notation into its three basic components to facilitate their analysis. The last notation is intended to be developed in the future into a representation based on optophonetic poetry. That is, a visual modification of a poem to evoke the oral performance of the reciter. Our proposal annotates the highlights of the performance to later be transformed into its final form. We provided examples of what the notations look like using the poem *Alto Jornal* by Spanish poet Claudio Rodríguez.

In the next section we described the tool itself. We did so by exploring its graphical interface and explaining each one of its elements. There, we explained the function of the different panels that conform the main window, as well as the functionality of every option of the different menus. We explained how the plots are interactive and how combined plots are internally related so that actions, like zooming and scrolling, applied to one of them reflect also on the other one.

In the use cases section, we presented common situations in which a user may use the tool. We used those scenarios as a walk-through of the software. We showed how to navigate the tool and work with it to obtain the desired output. This section is intended to be read not just as an example of the use of the tool, but also as a guide for new users.

Let us also explore the possibilities for future work and development. Perhaps the most important task at hand is detailing the optophonetic annotation. This is an interesting work because it expands the multidisciplinarity of the project. It would be crucial to involve graphic designers and experts in fields like colour theory. This would allow for a development of a critical and theory-based representation of the voice through the use of shape and colour of the words.

Further into the topic of annotations, the insight of expert pheneticists is crucial to perfect the phonetic analysis carried out by the tool. Not only on the algorithmic analysis of the data, but also on the annotations themselves. A view into particularities and nuisances of prosodic elements would be of great interest to achieve more polished annotations.

As stated previously, creating an ML model to improve this annotation process would be beneficial yet costly. Kitaev et al. (2018) shows that fine-tuning a model for a specific task and a specific language requires huge computational capacity. However, on a more promising note, it algo explores a way of combining pre-trained models to create a multilingual model. This reduces the cost and size of the resulting pre-trained model at the expense of an increased error rate. This has led to promising results such as (Kondratyuk and Straka, 2019), where the process was used to create a model for 75 languages by concatenating several pre-trained BERT models. This would allow to converge many poems in different languages to fine-tune a model. This would then allow for the study of poems in low-resources languages.

There are other interesting results than can further facilitate the application of resource-intensive techniques such as ML to an area that suffers from scarcity of adequate resources to use as input. Wang and Chang (2016) propose a graph-based bidirectional model that captures richer contextual information than other models. This allows to use fewer features than standard models. For our particular scenario, it could potentially be applied by using verses as if they were full poems. Then, the result of that would be fed to another classifier to put together the results of each individual verse.

Furthermore, the idea of applying ML to bolster the capabilities of the tool can have an extensive development. A large multimodal corpus consisting of results of the tool (i.e., the annotated texts) together with the original inputs, could be used to train or fine-tune a pre-trained model. This could help improve the precision of the tool´s annotations by substituting the heuristic algorithmics with the complex models of ML.

One of the more interesting aspects of using an ML model to boost the capabilities of the tool in the long run, is that it could be used to feed itself. The use of the tool will generate as output the annotated texts that could serve as input for a learning algorithm.

PAUSE being a computational tool gives it a high degree of flexibility that should be embraced if we want it to achieve its full potential. Proper maintenance of the software could also come in hand with its expansion according to user needs. A modular way of coding would allow the developers to offer a variety of combinations of different features from different notations. Allowing the user to put together or ignore some features would increase the flexibility, originality and depth of the analysis, bringing more value to the community. Access to parts of the code would also bring the possibility to programmatically alter aspects of the notations themselves, giving the users power over what, how and when to annotate.

On the same line, this tool is aimed at the research community. This means that all utilities and functionalities must be subject to the needs of the users. For that reason, this tool should be kept updated with new advances and standards of the field. At the same time, the very users of the software, through their use and observations, can provide with their expertise on the different disciplines touched upon this project. This expertise must be used to further improve the quality of the results of the tool. Similarly, all the notations proposed here must be polished and modified according to the results obtained by the users on many iterations that may show the weaknesses and strengths of our proposals. The usefulness of this project then must come from long-term interactions between developers and users.

Setting aside the possible technical improvements and enhancements, there are also a number of aspects that can be worked on in the future related to its functionalities and user experience.

One important functionality to allow for in depth studies of the poems' readings is to add video to the tool. This would bring the opportunity of studying the body language of the poet alongside their voice.

Another important functionality that should be expanded are the annotations and visualisations. More things should be added in the future coming from users' feedback and needs. These changes could range from tweaks to the existing annotations to new ones, or new ways of visualising the data.

When it comes to user experience, a web version could be introduced to avoid the need of downloading and installing software, to ease the maintenance and updates of the tool. On this note, it would be possible to use a cloud service. This would come with some added benefits. First, the ability to access one's work from different devices. Second, it would facilitate collaborative work by providing an easy way of sharing results with other researchers.

All in all, we believe that this project is a great addition to the current state of digital humanities and literary analysis and criticism. We think that it will be of interest for the research community and that it will be able to ease some aspects of future projects and studies. Even if it should currently be considered incomplete, we believe that it is a great starting point for a long and interesting project that will end up with a tool that can be of use for many different kinds of research. We have set the base to further theoretical studies and practical development, while proposing a tool that fulfils researcher needs that are not currently addressed.

7. Bibliography

- Aguilar, L., Roseano, P., Vanrell, M., de-la-Mota, C., Prieto, P. (2024) Sp_ToBI Traning Materials. Web page: Sp_ToBI Training Materials <https://sp-tobi.upf.edu>
- Alm, C., & Sproat, R. (2005) Emotional sequencing and development in fairy tales", in *Proceedings of the First International Conference on Affective Computing and Intelligent Interaction,* 668-674.
- Alm, C., & Sproat, R. (2005) Emotions from text: Machine learning for text-based emotion prediction, in *Proceedings of HLT/EMNLP*, 347-354.
- Apter, M. J. (1970). *The computer simulation of behaviour*. Hutchinson & Co.
- Baldick, C. (2008). *The Concise Oxford Dictionary of Literary Terms* (2nd ed.). Oxford University Press.
- Beckman, M., Díaz-Campos, M., McGory, J.T. & Morgan, T.A. (2002) Intonation across Spanish, in the Tones and Break Indices framework, *Probus*, 14(1).
- Beckman, M. E., Hirschberg, J. B., & Shattuck-Hufnagel, S. (2004) The Original ToBi System and the Evolution of the ToBi Framework, in Sun-Ah Jun (ed.), *Prosodic Typology: The Phonology of Intonation and Phrasing* (Oxford, 2005; online edn, Oxford Academic, 1 Feb. 2010).
- Bigi, B. (2016). A phonetization approach for the forced-alignment task in SPPAS, in *Human language technology: Challenges for computer science and linguistics (LNAI 9561)* (pp. 515-526). Springer.
- Boersma, P. (2001). Praat, a system for doing phonetics by computer, in *Glot International 5*(9/10), 341-345.
- Britannica, T. Editors of Encyclopaedia (2016). *caesura*. *Encyclopedia Britannica*.
- Burkov, A. (2019). *The hundred-page machine learning book.* Polen.
- Carnegie Mellon University. (2023). *CMU Sphinx: Open-source speech recognition*. https://cmusphinx.github.io/
- Cole, R., Mariani, J., Uszkoreit, H., Varile, G.B., Zaenen, A., Zampolli, Zue, V., eds. (1997). Survey of the state of the art in human language

technology*,* in *Natural Language Processing.* Vol. XII–XIII. Cambridge University Press

- Colonna, V. (2020). *Voices of Italian Poets: Analisi fonetica e storia della lettura della poesia italiana dagli anni Sessanta a oggi* [PhD thesis, Università degli studi di Genova, Università degli studi di Torino]
- Colonna, V. (2021). Voces de poetas. Introducción a un estudio fonético y empírico sobre Alberti, Guillén y Neruda, *in CHIMERA. Romance Corpora and Linguistic Studies,* 1, 109-136.
- Colonna, V., Pamies Bertrán, A., & Damato, S. (2024). Towards a phonetic history of the voices of Spanish poets: A first experimental study on the Generation of '27. *Estudios de Fonética Experimental, 33, 7–34.* Day, M., Dey, R. K., Baucum, M., Paek, E. J., Park, H., & Khojandi, A. (2021). Predicting severity in people with aphasia: A natural language processing and machine learning approach. *43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 2299-2302).
- Conlen, M. (n.d.). Kernel density estimation. *Mathisonian*. https://mathisonian.github.io/kde/
- Dasgupta, S., Papadimitriou, C. H., Vazirani, U. (2006). *Algorithms* (1st ed.). McGraw-Hill.
- Delmonte, R., & Prati, A. (2014). SPARSAR: An Expressive Poetry Reader, in *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 73-76.
- Elvira-García, W., Roseano, P., Fernández-Planas, A. M., Martinez-Celdran, E. (2016). A tool for automatic transcription of intonation: Eti_ToBI a ToBI transcriber for Spanish and Catalan, in *Language Resources and Evaluation*, 50(4), 767-792.
- Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J. (2019) A guide to deep learning in healthcare. *Nat Med*. 25(1):24-29.

- Estebas-Vilaplana, E.; Prieto, P. (2009) La notación prosódica en español. Una revisión del Sp_ToBI, *Estudios de Fonética Experimental XVIII*, 263-283.
- Estebas-Vilaplana, E., Prieto, P. (2010). Castilian Spanish intonation. In: Prieto, P. & Paolo Roseano (eds.) (2010) *Transcription of Intonation of the Spanish Language*, LINCOM Studies in Phonetics 06
- Face, T.; Prieto, P. (2007) Rising accents in Castilian Spanish: a revision of Sp_ToBI, *Journal of Portuguese Linguistics*, 6.1, 117-146.
- Hossain, S. (2019). *Visualization of bioinformatics data with Dash Bio*. In C. Callaway, D. Lippa, D. Niederhut, & D. Shupe (Eds.), *Proceedings of the 18th Python in Science Conference* (pp. 126-133).
- Hossain, M.T., Rahman Talukder, M.A., Jahan, N. (2021). Social networking sites data analysis using NLP and ML to predict depression. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*.
- Hunter, J.D. (2007). Matplotlib: A 2D graphics environment, in *Computing in Science & Engineering*, 9(3), 90–95.
- Jadoul, Y., Thompson, B., de Boer, B. (2018). Introducing Parselmouth: A Python interface to Praat, in *Journal of Phonetics*, 71, 1-15.
- Kitaev, N., Cao, S., Klein, D. (2018). Multilingual constituency parsing with self-attention and pre-training, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Association for Computational Linguistics.
- Katsamanis, A., Georgiou, P.G., Goldstein, L.M., Arayanan, S.N. (2011). *SailAlign: Robust long speech-text alignment*.
- Kondratyuk, D., Straka, M. (2019). 75 Languages, 1 Model: Parsing Universal Dependencies Universally, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2779–2795. Association for Computational Linguistics.
- Koreman, J. (2006). Perceived speech rate: The effects of articulation rate and speaking style in spontaneous speech. *J Acoust Soc Am,* 119(1):582-96.

- Li, J., Meng, Y., Wu, Z., Meng, H., Tian, Q., Wang, Y., Wang, Y. (2022). Neufa: Neural network based end-to-end forced alignment with bidirectional attention mechanism, in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8007-8011).
- Lu, Z., Sim, J., Wang, J.X., Forrest, C.B., Krull, K.R., Srivastava, D., Hudson, M.M., Robison, L.L., Baker, J.N., Huang, I.C. (2021). Natural Language Processing and Machine Learning Methods to Characterize Unstructured Patient-Reported Outcomes: Validation Study. *J Med Internet Res* 23(11).
- Lubbers, M., Torreira, F. (2013-2018). *Praatalign: An interactive Praat plug-in for performing phonetic forced alignment* (Version 2.0). GitHub. https://github.com/dopefishh/praatalign
- Mahdi, O.A., Mohammed, M.A., Mohamed, A.J. (2012). Implementing a novel approach to convert audio compression to text coding via hybrid technique, in *International Journal of Computer Science Issues, 9*(6), 53–59.
- Mishra, S. K., Kundarapu, H., Saini, N., Saha, S., & Bhattacharyya, P. (2020). IITP-AI-NLP-ML@ CL-SciSumm 2020, CL-LaySumm 2020, LongSumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing* (pp. 270-276). Association for Computational Linguistics.
- Mistrorigo, A. (2018). *Uso crítico de las grabaciones: La voz de los poetas, uso crítico de sus grabaciones y entrevistas.*
- Moreno, P. J., Joerg, C. F., Van Thong, J. M., Glickman, O. (1998, December). A recursive algorithm for the forced alignment of very long audio segments. In *ICSLP* (Vol. 98, pp. 2711-2714).
- Nowell Smith, D. (2015). *On Voice in Poetry: The Work of Animation*. (Language Discourse Society). Palgrave.
- Ong, C.J., Orfanoudaki, A., Zhang, R., Caprasse, F.P.M., Hutch, M., Ma, L., et al. (2020) Machine learning and natural language processing methods to identify ischemic stroke, acuity and location from radiology reports. *PLoS ONE,* 15(6).

- Osorio, J., & Beltran, A. (2020) Enhancing the Detection of Criminal Organizations in Mexico using ML and NLP, in *International Joint Conference on Neural Networks (IJCNN*, 1-7).
- Panayiotou, A., Bon, A. (2023). *Overcoming complex speech scenarios in audio cleaning for voice-to-text.*
- Plotly Technologies Inc. (2015). *Collaborative data science*. Plotly Technologies Inc. https://plot.ly
- Plug, L., Lennon, R., & Smith, R. (2022) Measured and perceived speech tempo: Comparing canonical and surface articulation rates, in *Journal of Phonetics*, Volume 95,101193.
- Python Software Foundation. (2020). *Sunsetting Python 2*. https://www.python.org/doc/sunset-python-2/
- Quinn, J. (2020). Dive into deep learning: tools for engagement*.*
- Rahul, S. A., & Monika. (2020). NLP-based machine learning approaches for text summarization. *Fourth International Conference on Computing Methodologies and Communication (ICCMC)*.
- Read Beyond. (2017). *Aeneas*. https://www.readbeyond.it/aeneas/
- Reddy, M. (2011). *API design for C++*. Elsevier Science.
- Robert, J., Webbie, M., et al. (2018). *Pydub*. GitHub. http://pydub.com/
- Saxena, A. K. (2022). Enhancing Data Anonymization: A Semantic K-Anonymity Framework with ML and NLP Integration. *Sage Science Review of Applied Machine Learning*, *5*(2), 81–92.
- Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., & Hirschberg, J. (1992). TOBI: A standard for labeling English prosody. In *International Conference on Spoken Language Processing* (pp. 867–870). Banff, Canada.
- Simonoff, J.S. (1996). *Smoothing methods in statistics* (1st ed.). Springer.
- Steffensen, J.F. (1950). *Interpolation* (Second edition). Dover Publications.
- Wang, W., & Chang, B. (2016). Graph-based Dependency Parsing with Bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315. Association for Computational Linguistics.

- Waskom, M.L., (2021). seaborn: statistical data visualization, in *Journal of Open Source Software*, 6(60), 3021.
- Wolfram, S. (2002). *A new kind of science*. Wolfram Media, Inc.
- Zhou, Z. H. (2021). *Machine learning*. Springer nature.