

Movie Analyzer & Player using the ‘two clocks’ method

Pre – Report

Dean Shandler 302221270

Kirill Privalov 323650119

Instructors:

Zvi Lotker

Yoram Segal

Dror Ravitz

Project Number:

Student Signature: \_\_\_\_\_\_\_\_\_\_\_ Prof Signature: \_\_\_\_\_\_\_\_\_\_\_

Table of Contents

**Type chapter title (level 1)1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

1. Abstract

This project implements the idea of using two clocks to analyze plays and movie scripts. An idea represented in “The Tale of Two Clocks” by Zvi Lotker. As shown in Zvi’s article, the main idea of the Two Clocks algorithm is to identify critical events in a given relationship network using the drift between two clocks.

We’ll analyze movie scripts and their subtitles to output the exact moments of critical events in the given movie and represent these moments in a custom Media Player. The viewer will be able to select these crucial moments in the film and observe them.

The goal is that this project may be used as a research tool for analyzing multiple movies for finding patterns within them.

* 1. Key words

Two Clocks, Movie, Movie Script, Subtitles, Media Player, Critical Moments,

1. Introduction and motivation

This project can help build a database of movie patterns that can be analyzed in the future using Machine learning techniques for identifying common patterns between different movies.

This can be extended upon to suggest movies of certain interests or preferences to viewers in streaming platforms such as Hulu, Netflix, HBO and Amazon Prime.

Critical moments in the movies can be used to build a “Summary Trailer” that display these key moments.

1. The Problem

We have succeeded in analyzing movie scripts using the “Two Clocks” method presented in Zvi’s article. But this analysis is limited to the relationship network contained only in the script. The challenge we face is how to represent these crucial moments in real time, so they can be represented while watching the movie.

A movie script is timeless, and holds descriptions of feelings, emotions, and general descriptions of the scenes.

To overcome the challenge of representing time we use the subtitles of the movie that contain time-stamps of the sentences said in the movie. Usually, the published script isn’t a fully matched with the published version of the movie hence it is not matched completely with the subtitles text. Here we face a challenge of finding a suitable match between a critical event to the actual time it happened in the movie.

Finding the critical moments without taking into consideration the descriptions present in the script, we must clean the script of these descriptions. Our current

limitation is that we need the script to be symmetrical hence, the scene descriptions, spoken words and name of speaker must be in the same column.

1. Project Goals

* Finding Critical moments in a film, represented by corresponding timestamps.
* Creating bookmarks of these moments and showing them in a media player while the film is playing.
* Creating a database of movies and their critical moments for further analyses and pattern recognition.

1. Tools and Software
   1. Python and PyCharm IDE:

We have decided to program the projects algorithms and routines with the Python scripting language using the PyCharm community IDE.

* 1. Java Programming language:

We have decided to program the Media Player with the Java programming language.

1. Theory Background

**The definition of a clock in the algorithm** – A clock C is a strictly monotonic bijective continuous function from the standard time interval [τs, τe] to the measured time interval [µs, µe], i.e., C : [τs, τe] → [µs, µe]. The domain of a clock is called the standard time, and the range of a clock is called the measure time.

**There are two types of clocks** – Event clock - is the reflection of the structure of conversation in the movie.

weighted clock – captures the complexity of what is being said.

**Choosing clocks for our project** – Event clock – Speaker change in the script.

weighted clock – The number of words spoken until a certain time n.

**The algorithm takes the input of two Normalized clocks** – A clock C would be called normal if its domain and range are the unit interval i.e., C : [0, 1] → [0, 1]

**Manipulation done on Normalized Clocks** – We create a third vector called Dynamic Clock that represents dynamics between the two clocks. The vector is calculated as follows:

Dynamic\_Clock[i] = Normalized\_Clock1[i] – Normalized\_Clock2[i]

**Plotting a graph** – We derive a graph that represents the change in the dynamics between the two original clocks in the time interval [0, 1] by defining the X axis as the Normalized\_Clock1 and the Y axis as the Dynamic\_Clock.

**Examples of Graphs** – The next three graphs represent the dynamics between the weighted clock and event clock in the plays Julius Caesar, Othello and Romeo and Juliet.

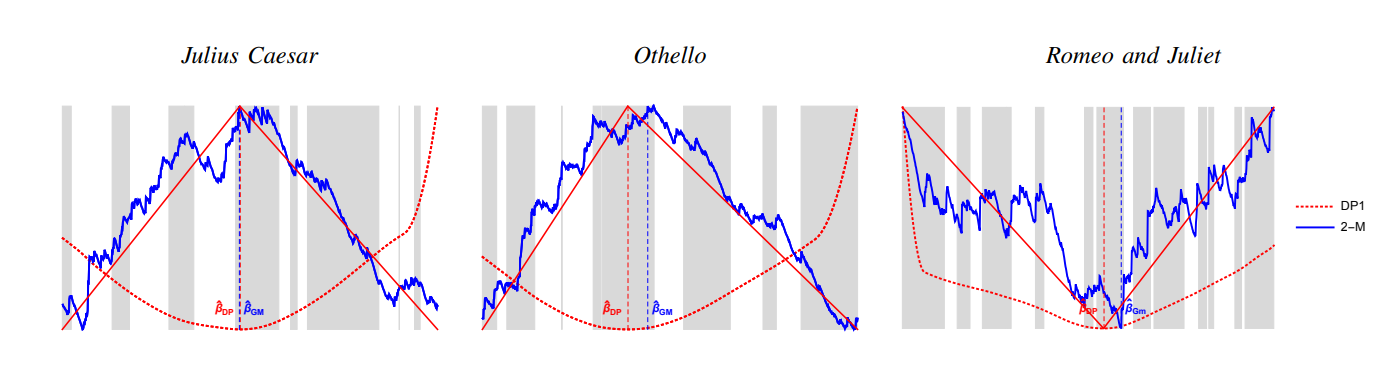


Figure 1 Examples of graphs from article

1. Method description

**Part 1 – Two Clocks Algorithm – Finding critical Timestamps**

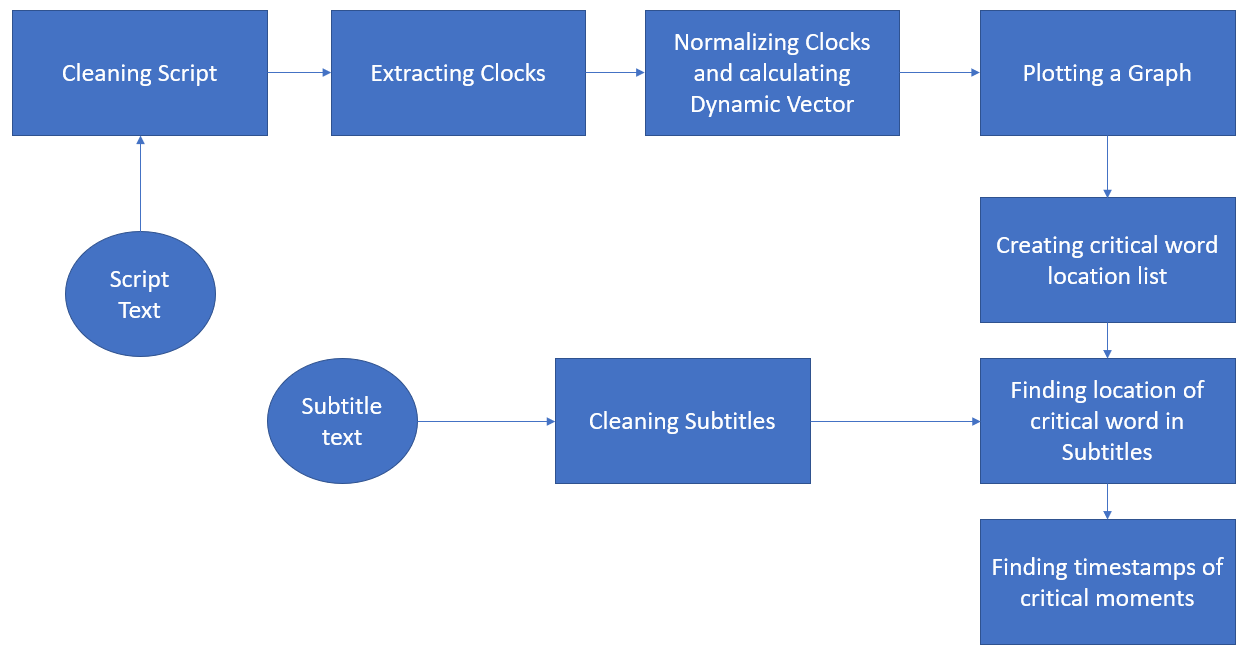


Figure 2 Project modules

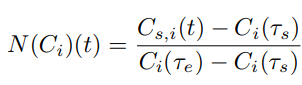
Module explanation:

**Cleaning Script:** receives a path to a script.txt file. the function cleans the script of all words that are not - who spoke them or what they said, this removes the scene descriptions and emotions from the script. We use regular expressions to clear any instances of parentheses and brackets within the text which usually describe emotions and actions.

**Cleaning Subtitles:** receives a to the subtitles text file and removes all parentheses and brackets using a Regular Expressions.

**Extracting Clocks:** Receives the clean Script and creates a table with two columns that contain the two clocks. First column – sums up the number of times a speaker has changed. Second column – sums the number of words spoken to a certain point.

**Normalizing Clocks and calculating Dynamic Vector:** This function receives a path to the table of clocks Clk1 and Clk2. then creates a new file that contains the normalized clocks: N(Clk1) & N(Clk2) with this formula:

****

Then calculates the Dynamic\_Vector in column 3 (N(Clk1) – N(Clk2)).

**Plotting a Graph:** Graph plot for X-Axis: N(Clk1) and Y-Axis: Dynamic\_Vector

**Creating critical word location list**: This Function calculates the derivative of normal Clocks, so we can find the critical moments (Minimum and Maximum points in the graph) This is done by using this formula:

Derivative[i] = Dynamic\_Vector [i] - Dynamic\_Vector [ i+1]

We also find the N largest and the N smallest derivatives for timestamps.

**Finding location of critical word in Subtitles:** This function receives the critical word location list and searches for a match between the script and the subtitles using sequence matching of the words and returns a list containing the indexes of the words in the subtitle file.

**Finding timestamps of critical moments:** This function receives a list containing the indexes of the critical words in the subtitle file, then returns their corresponding Timestamps.

**Part 2 – Media Player**

**Part 3 – Database**

1. Completed Tasks
2. Schedule and Phases
3. Technical Information
4. References
5. Table of figures