Poetry Analysis Using NLP

Hung Trinh
Computer Science and Engineering
University of Oulu
Oulu, Finland
Email: Hung.TRINH@student.oulu.fi

GIT LINK

You can find the project documentation, its source code, and usage instruction on Github at this link: https://github.com/manhhungking/PoetryAnalysisNLP

Abstract* — *In this paper, we study automatic analysis of poetry and highlight natural language processing (NLP) approaches—using the Natural Language Toolkit (NLTK) for Python—that identify key poetic features such as style, rhyming schemes, and emotional tone. This research explores structural and semantic patterns within Shakespear Sonnets from the vantage of elements like word frequency, repetition and named entities. SentiWordNet, plays a role in measuring the mere sentiment expressed and then lexicon-driven emotion analysis helps further comprehend. This approach aims to delimit a strong foundation for understanding the rich linguistic and stylistic characteristics of poetry using NLP, providing insights into classical literature in a scalable fashion.

Index Terms* — *Poetry Analysis, Natural Language Processing (NLP), Sentiment Analysis, Emotion Detection Rhyme Analysis, Shakespeare Sonnets, Lexical Frequency, Named Entity Recognition, SentiWordNet, EmoTag1200, Semantic Analysis, NLTK

1 Introduction

Poetry is a unique and powerful form of writing that communicates multiple layers of meaning, structure, and emotion. There are, as always, difficulties and opportunities in looking at poetry this way — that is to say, by distributing the labor of reading across a computerized network. It allows us to discover trends, emotions and aesthetics that we may overlook. This burst of interest in building datasets and practice for literary study has led to collaborative volumes exploring the challenges and opportunities afforded by NLP tools. This even enables researchers to quantify and demonstrate aspects of poems such as emotion, sentiment, rhythm and rhyme.

I analyze style and sentiment in poetry, focusing primarily on the sonnets of Shakespeare: In this study we apply natural language processing (NLP) techniques to extract various metrics for measuring both stylistic features as well as emotional states. The clear structure of Shakespeare's sonnets and their variety in theme make them a perfect measurement for computers. This project focused on some of thesubtle and high level aspects in analyzing poems. This analysis looks at things like the types of words that are used, how frequently words are repeated, identifying people or place names, interpreting emotion in poetry, and quantifying the level of emotion.

I employ Python NLTK library with some additional tools to achieve these goals such as SentiWordNet that helps to understand the feelings and EmoTag1200 for emotion annotation. These tools can assist us with measuring things such as general sentiments, emotional ratings based on the words used in tweets, and rhyming patterns. You can also visualize those results with various types of charts and graphs. Building on these poetic elements, we put our feelings in poetry on a two dimensional map and examine how feeling(s) and stylistic devices work together to create meaning (what is going on) and mood (how it feels to you or the speaker of the poem).

These studies reveal the potential of computational approaches to studying extracts from poetry. It guides users in exploring how language interacts with feeling and style, and form in poetry. It also demonstrates how the automated text analysis capabilities in natural language processing (NLP) can be applied to analyze a large corpus of human-written text useful for researchers in digital humanities and computational linguistics related fields.

2 Literature Review

2.1 Traditional Methods in Poetry Analysis

Traditionally, poetry analysis has been a close reading enterprise, where the scholar closely examines various aspects (e.g., meter and rhyme; style) to provide an interpretation. If we consider the metrical side, Fabb and Halle [1] provide a detailed approach to English meter, addressing the contribution of such properties to the rhythmical and interpretative layers of literary language. Traditional methods, based on qualitative readings of rhythm and structure (and no small degree of subjective literary judgement), rely on their paradigm for recognizing metric patterns.

Manual annotation also played a key role in linguistic studies where linguists annotated lexical choices, syntactic constructions and patterns found in the text. Biber and Conrad [2] address these general patterns in their research on genre and style, describing how features of language associated with a given genre may contribute to meaning. Such features are what allows texts of poetry to hold different layers of depth and resonance than any other form of literature.

2.2 Recent Advances in Computational Poetry Analysis

Advancements, in corpus linguistics and natural language processing (NLP) have become crucial in the analysis of poetry according to McEnery and Hardie [3]. They emphasize that corpus linguistics enables the study of text collections for an assessment of linguistic features, across a vast array of literary works. Bird, Klein, and Loper's [4] description of the development of NLP tools has enabled scholars to automate tasks, like tokenization and identifying parts of speech. Named entities, in complex poetic texts.

Moreover in times sentiment analysis tools have seen advancements in their ability to recognize emotional nuances, within poetry works. This is highlighted by Liu [5] who discussed methods for extracting sentiments from text by focusing on the language found in works. In our research project we make use of SentiWordNet 3.O [6] to evaluate positive and negative sentiment levels for each line of poetry. This approach offered an evaluation of sentiment distribution that complements interpretive methods. SentiWordNet 3.O is a resource with a lexicon that links words to sentiment values enabling a more precise identification of sentiments, in various poetic excerpts.

2.3 Comparision of of Traditional and Computational Approaches

Traditional and computational approaches both have unique strengths when it comes to poetry analysis. Conventional techniques provide subtle interpretative insights that are capable of capturing implicit meaning based on context that may be missed by automated processes. For example, traditional meter analysis as described by Fabb and Halle [1] involves knowledge of cultural and historical contexts that computational models lack a priori.

On the other hand, computational approaches provide scalability and accuracy to analyze large text data that otherwise is impractical for hand coding. An example of this can be seen in the Stanford Transhistorical Poetry Project [7] which indicates how computational methods can work with poetic datasets spanning more than one time period allowing for an investigation into features such as rhyme schemes that cut across different eras of poetry. Both methods are able to reveal forms of poetic structure that remain stable or change over the course of history, demonstrating the power of large scale corpus analysis as a tool for literary historical study.

And further, recent computational work relying on advances in computational techniques (mostly embedding-based approaches for stylistic and emotional analysis) start to move away from superficial linguistic analysis. Embeddings enable researchers to represent semantic relationships between words in multi-dimensional space, thus making latent themes and stylistic tendencies possible to detect with easy access. Embeddings can also shed light on how words with similar meanings group together, providing an interesting opportunity to explore themes and motifs in poetry analysis.

2.4 Applications of Emotion and Sentiment Analysis in Poetry

This project focuses on one key objective, specifically giving empirical measurement and visualization of the affect in Shakespeare's sonnets. I establish a middle ground between qualitative reading and quantitative analysis by examining poetic language in terms of its emotional or affective qualities through the lens of emotion and sentiment analysis tools. As described by Liu [5], we can score each line's sentiment using SentiWordNet [6] to identify where the positive and negative tones are. This gives a three-dimensional nature to the way Shakespearean language can be understood, particularly with each emotion (for example melancholy or joy) clearly mapped through what words are chosen.

Furthermore, we utilize EmoTag1200 lexicon [8] in order to map emotions to four dimensions: anger joy fear and surprise. This project creates lines, which are displayed to represent the emotional movement throughout a sonnet by being positioned in some 2D space of emotions. These visualisations provide insights into emotional intensity behaviour and support comparative analysis between sections of a poem, thus augmenting traditional close reading with an element of the quantitative.

2.5 Integrating Computational Methods with Traditional Interpretation

Computational tools allow large-scale analysis and objectivity. But they work best as part of a two-step process, where such big-data computation is organized around traditional interpretative frameworks. As one approach, we could cluster lines with the most extreme sentiment scores based on an initial examination of sentiment scores across our text to guide selection and use close reading methods to interpret the context for these lines. This combined method helps us to get the general structure of Shakespeare Sonnets and minor variations that are peculiar of poetry language.

Combining traditional and computational approaches, this project will offer a systematic analysis of key poetic features such as form (e.g. style/rhyme), sentiment, and emotion. The combination of close reading and quantitative analysis not only improves the quality of the results, but also showcases how valuable NLP methods can be to research in literary studies, opening a door to new interpretations of time-honored poetry.

3 Methodology

In this section, I will discuss the methods used to explore the selected poetry. Focusing primarily on the computational techniques, we can analyze various linguistic features such as sentiment, emotion, meter and rhyme. This method builds

off of traditional close reading techniques, yet also employs modern-day natural language processing (NLP) tools to supply both qualitative understanding and quantitative analysis of the Shakespeare Sonnets.

3.1 Data Collection and Preprocessing

I download Shakespeare Sonnets [9] from a publically available corpus. This dataset is prepared to make it suitable for nlP analysis Preprocessing included tokenization which split the sonnets by words and lines; removal of stopwords; lemmatization which aspired to convert as many words into their root forms. Just keep in mind that there are few steps where we make use of the original poem to maintain its originality. It is downloaded to the same working directory and it will be later used in analysis.

3.2 Poetry Analysis Tasks

I broke the project down into types of tasks that are aimed at analysing some different aspects of poetry. The following are the different NLP techniques used to accomplish these tasks.

3.2.1 Part-of-Speech (POS) Tagging

The analysis of synactic structure in the sonnets is done by performing part-of-speech tagging on each line using Python's Natural Language Toolkit (NLTK) library [4]. In the text every POS tag is assigned to each token, allowing us to measure specific syntactic components of English. Next, we created a visualization that shows the frequency distribution of POS tags in order to gain insight into stylistic tendencies (for example, on how nouns, verbs, adjectives are varied throughout different parts of the poem).

3.2.2 Named Entity Recognition (NER) Analysis

Regarding the Named Entity Recognition (NER) results, I extracted and studied named entities — where each entity denoted a unique object in the datasets, such as persons, locations, organizations based on Spacy's NER tool. I charted the frequency of each entity type and created other visualizations pointing to those lines where we found these entities. This analysis highlights the facility of character and place references that provide ways into a thematic architecture for the sonnets.

3.2.3 Sentiment Analysis

I measured the emotional tone of each sonnet applying sentiment analysis with SentiWordNet 3.0 [6]. The positivity of the negative score of each line are computed. And then they are plotted on a 2D grid to visualize them.

3.2.4 Emotion Analysis

The EmoTag1200 lexicon [8] is then utilized to tag sonnet words over a wide variety of emotions, including anger, joy, surprise and sadness. I then mapped this lexicon onto the lines of the sonnet, which ultimately gave us a way to plot how emotions change over the course of the text.

3.2.5 Rhyme and Meter Analysis

Stanford Transhistorical Poetry Project [7] played a important role in analysing structure of the sonnets, namely rhyme scheme and meter. Combined with some basic custom written scripts, we used the poesy library to find rhyming pairs and tag them based on their rhyme scheme and compare it against classical sonnet forms. This also produced counts of syllables for each line, providing a better comprehension of how the meter are applied to achieve the rhythm and tone in each poem.

3.3 Tools and Libraries

This study is performed with the following Python libraries and tools:

- NLTK (Natural Language Toolkit) [4] is used for tokenization, part-of-speech tagging, and lemmatization.
- SentiWordNet [6] is used to perform sentiment analysis.
- EmoTag1200 [8] is employed to extract emotion-related features from the text.
- Stanford Transhistorical Poetry Project [7] is used for rhyme and syllable structure analysis.
- Matplotlib and Seaborn are employed for visualizing the sentiment and emotion distribution.

4 Implementation

This section describes the implementation of each task in Project 10: Poetry Analysis Using NLP. .

4.1 Task 1: Part-of-Speech Tagging and Frequency Analysis

In this task, we tag each token in the sonnet with a part-of-speech (POS) label and analyze the frequency of different POS tags.

Algorithm 1 Part-of-Speech Tagging and Frequency Distribution

```
Input: Raw Shakespeare Sonnets
Output: Frequency distribution of POS tags in the poem
for each line in Sonnet do
    Apply POS tagging to the tokens of a line
    for each token with its tag do
        Count the distinct tag
        Increment the count for the POS tag
    end for
end for
Plot the bar chart of POS tag frequencies and graph for word positions
```

The POS tags of each token are determined using an NLTK POS tagger, and the frequency distribution of tags is visualized using a bar chart.

4.2 Task 2: Frequency of Top 30 Terms and Their POS Tags

This task identifies the 30 most frequent terms in the sonnets along with their POS tags and their position in the

lines.

Algorithm 2 Extract Top 30 Terms and Their POS Tags

Input: Preprocessed Sonnets
Output: Top 30 terms, their POS tags, and positions
Initialize an empty dictionary term_freq
Tokenize the Sonnets into words
Apply POS tagging to the tokens
for each token with its tag **do**
    Update term_freq with the frequency of the token
end for
Sort term_freq in descending order
Select top 30 terms and their POS tags
Plot the frequency of the top 30 terms with their positions in the poem

This algorithm identifies the most frequent terms in the sonnets, and stores their respective POS tags and positions for visualization.

4.3 Task 3: Repetition Detection

This task evaluates the number of repetitions occurring within the same line or between successive lines of the poem by counting the number of repeated tokens per consecutive lines and calculating the distribution of repetitions.

Algorithm 3 Detect Repetitions in Successive Lines

Input: Raw Sonnets
Output: Statistics on repetitions in successive lines
for each pair of successive lines **do**
    Initialize repetition_count = 0
    for each token in line 1 **do**
        if token appears in line 2 **then**
            Increase repetition_count by 1
        end if
    end for
    Store repetition_count for the line pair to an array
end for
Calculate mean, standard deviation, and kurtosis of repetition counts
Plot the distribution of repetitions

4.4 Task 4: Named-Entity Recognition

I use Spacy's named-entity tagger to identify named entities (person, organization, location) in each line of the poem. I also provides a frequency and occurence analysis then visualize them using histogram.

Algorithm 4 Named-Entity Recognition in the Poem

Input: Raw Sonnets
Output: Frequency and occurence of named entities in the poem
for each line in Sonnet **do**
    Apply SpaCy named-entity recognition to the line
    for each entity in the line **do**
        if entity is of type Person, Organization, or Location **then**
            Increment the count for the entity type
            Record the line number where the entity appears
        end if
    end for
end for
Plot a histogram showing the frequency of each named-entity type
Plot the line numbers where each type of named entity occurred

4.5 Task 5: Emotion Analysis Using EmoTag1200

In this task, we use the EmoTag1200 lexicon to detect emotions in words that appear in the lexicon and represent the emotion of them as points in a 2D space.

Algorithm 5 Emotion Analysis Using EmoTag1200 Lexicon

Input: Raw Sonnets, EmoTag1200 lexicon
Output: Emotion coordinates for each word (anger, joy) and (fear, surprise)
for each line in Sonnet **do**
    for each word in line **do**
        Retrieve the emotion scores (anger, joy) and (fear, surprise) from EmoTag1200
        Store the word's emotion coordinates in a 2D space
    end for
end for
Plot emotion coordinates (anger, joy) and (fear, surprise) for each word

4.6 Task 6: Emotion Analysis for Each Line

This task is similar to task 5. However, we consider analyzing emotion of each line instead of word level. In detail, we compute the total emotion of each line by summing the emotion scores of words in the line then print out the overall emotion for each line.

Algorithm 6 Emotion Analysis for Each Line of the Poem

Input: Raw Sonnets, EmoTag1200 lexicon
Output: Total emotion score for each line
for each line in Sonnet **do**
    Initialize total_emotion = (0, 0) for (anger, joy) and (fear, surprise)
    for each word in line **do**
        Add the emotion score of the word to total_emotion
    end for
    Store total_emotion for the line
end for
Print out total emotion score for each line

4.7 Task 7: Sentiment Analysis Using SentiWordNet

For sentiment analysis, we use SentiWordNet to calculate the overall sentiment of each line by averaging the sentiment scores of individual tokens.

Algorithm 7 Sentiment Analysis Using SentiWordNet

Input: Raw Sonnets, SentiWordNet
Output: Sentiment score of each line
Initialize sentiment_score = 0
for each word in line **do**
    Retrieve sentiment score from SentiWordNet of the word
    Add sentiment score to line_sentiment_score
end for
Store line_sentiment_score
Plot the sentiment scores for each line (positive vs negative)

4.8 Task 8: Exploring Key-Location Entities with Emotion and Sentiment

In this task, I explore key-location entities and their associated emotions and sentiments using Principal Component Analysis (PCA) to represent emotion vectors.

Algorithm 8 Exploring Emotion and Sentiment Around Key-Location Entities

Input: Raw Sonnets, top 2 location entities, EmoTag1200 lexicon
Output: PCA plot of emotion vectors around location entities
for each key-location entity in sonnets **do**
    for each instance of the entity in the poem **do**
        Extract a window of two tokens around the entity
        Identify emotion words in the window using EmoTag1200
        Calculate the average emotion vector for the window
    end for
    Apply PCA to reduce the 8D emotion vector to 2D
    Store the PCA result for visualization
end for
Plot the emotion vectors for each location entity using PCA

4.9 Task 9: Rhyme and Syllable Structure Analysis

I analyze rhyme and syllable structure using the poesy library and identify the most frequent subsequences of rhyme patterns.

Algorithm 9 Rhyme and Syllable Structure Analysis

Input: Raw Sonnets, **poesy** library
Output: Rhyme sequence and frequent subsequences
Extract the highest possible rhyme of the whole poem using poesy
Identify the most frequent subsequences in the rhyme_sequence
Calculate the length of each subsequence and their frequency
Print the subsequence lengths and their frequencies

4.10 Task 10: Literature Review and Findings

In the final task, I summarize the results from earlier tasks, discuss them in relation to corpus linguistics literature and point out some methodological limitations for our data processing pipeline.

The following components are addressed in Task 10:

1. Analysis of Findings: The results obtained from each task are analysed according to their significance, impact on knowing the properties of Shakespearean sonnets.
2. Comparison with Literature: I compare the result with previous research in corpus linguistics.
3. Methodology Gaps: Any observed gaps or challenges in the methodology are identified, particularly in the sentiment and emotion analysis. Suggestions are made to enhance the robustness of these analyses.
4. Limitations and Future Work: I discuss the limitations of our data processing pipeline and recommend improvements for future work.

5 Result

5.1 Task 1: Part-of-Speech (POS) Tagging Distribution

Using NLTK's POS tagger, each token in the text is tagged, and the distinct POS tags are counted per line. Figure 1 shows a bar plot of the frequencies of POS tags across the poem:

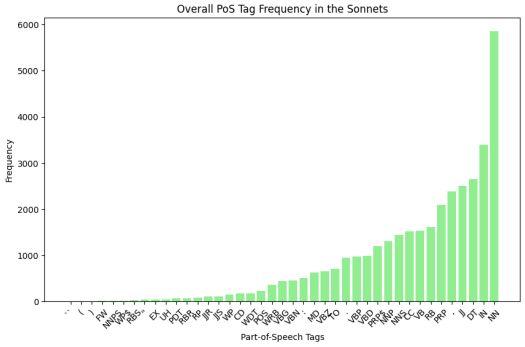

Fig. 1: PoS Tag Frequency of the Sonnets

- The plot shows that nouns (tagged as NN) and prepositions (tagged as IN) are the most common categories in all tags. The number of nouns is almost double the share of prepositions, suggesting that the sonnets put an emphasis on naming or descriptive words.
- Adjectives (JJ), though they are less frequent than the other two but still high in frequency, which ends up giving depth and emotional nuance to the visualisation.

5.2 Task 2: Frequency of Common Terms and Their Positions

This task identifies the 30 most common terms, their POS tags, and their positions in each line. Figure 2 visualizes these terms, with color representing their positions:

- Key terms such as "love" and "time" appear prominently, often positioned near the middle or end of lines for emphasis. This deliberate positioning suggests that poets are carefully structuring impactful words to maximize their emotional resonance.
- The frequency and placement patterns reinforce thematic elements and highlight the poet's use of repetition and strategic word choice.

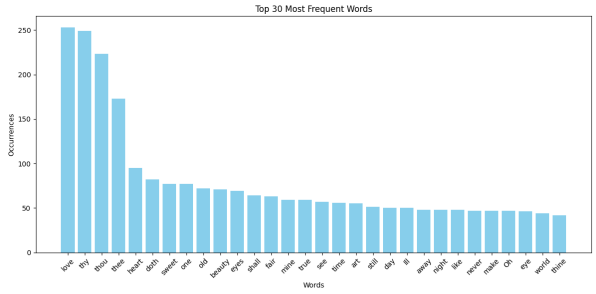

Fig. 2: Frequency of top 30 frequent words

Figure 3 shows the top 30 highest repetitive words with their number of occurrences and PoS Tag:

	Word	Occurrences	POS Tag
0	love	253	VBP
1	thy	249	JJ
2	thou	223	NN
3	thee	173	NN
4	heart	95	NN
5	doth	82	NN
6	sweet	77	JJ
7	one	77	CD
8	old	72	JJ
9	beauty	71	NN
10	eyes	69	NNS
11	shall	64	MD
12	fair	63	JJ
13	mine	59	NN
14	true	59	JJ
15	see	57	NN
16	time	56	NN
17	art	55	NN
18	still	51	RB
19	day	50	NN
20	Ill	50	NN
21	away	48	RB
22	night	48	NN
23	like	48	IN
24	never	47	RB
25	make	47	VBP
26	Oh	47	NNP
27	eye	46	NN
28	world	44	NN
29	thine	42	JJ

Fig. 3: Occurence and PoS tag of top 30 repetitive words in the Sonnets

Figure 4 highlights the word postition in the line:

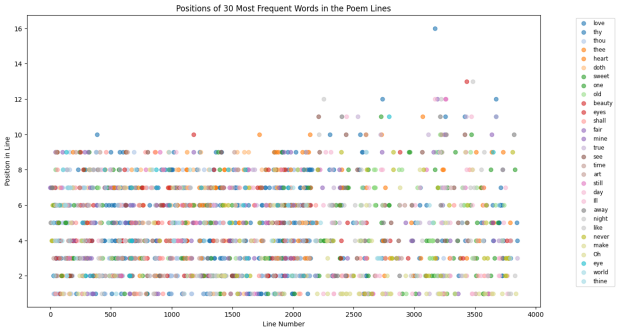

Fig. 4: Word position of top 30 frequent words

5.3 Task 3: Repetition Analysis

In this task, repetitions within and across lines are analyzed. Figure 5 shows statistical value of repititive words per successive lines. Figure 6 shows a distribution plot of the repetitions.

```
Mean repetitions per two-successive lines: 0.7411642411642412
Standard deviation of repetitions: 1.0375235869121533
Kurtosis of repetitions: 7.00575305461434
```

Fig. 5: Word position of top 30 frequent words

- Mean repetitions is approximately 0.74. This indicates that on average, words in two successive lines are repeated 0.74 times. This is not a significant number, but it indicates that the repetitions are not extremely rare or abundant.
- Standard deviation of repetitions is approximately 1.04. This shows a moderate level of variation in repetition count across line pairs.
- Kurtosis of repetitions is approximately 7. Because 7 ¿ 3, so the data has heavy tails or more extreme outliers than a normal distribution. This means that most line pairs have low or zero repetitions, but there are occasional line pairs with significantly higher counts, contributing to the ”heavy tails” in the distribution.
- Observing the plot in Figure 6, it is tendential to come up with same word in two lines next to each other as mean and standard deviation show a normal distribution, which indicates that a poet regularly emphasis some terms via repetition.

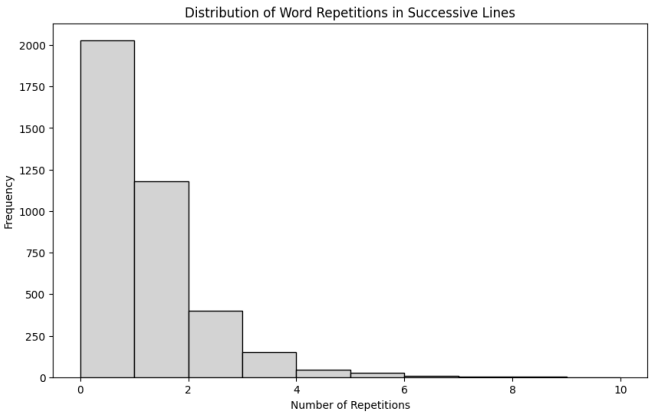

Fig. 6: Word position of top 30 frequent words

5.4 Task 4: Named Entity Recognition (NER)

I make use of Spacy to perform entity extraction and visualization in the text. As such, a histogram marking the frequencies of the respective entity types is shown in Figure 7 and line numbers highlighte where certain entities are present by Figure 8:

- Entities that express people are the most common entity type, capturing their symbolic and narrative importance, followed by locations with additional context embedded in them.
- The line-by-line plot in Figure 8 shows concentrated groups of mentions to the same entity, meaning there are symbolic reference points to people/places that overlap more with sections of text, which might denote important poetic themes or changes in tonality.

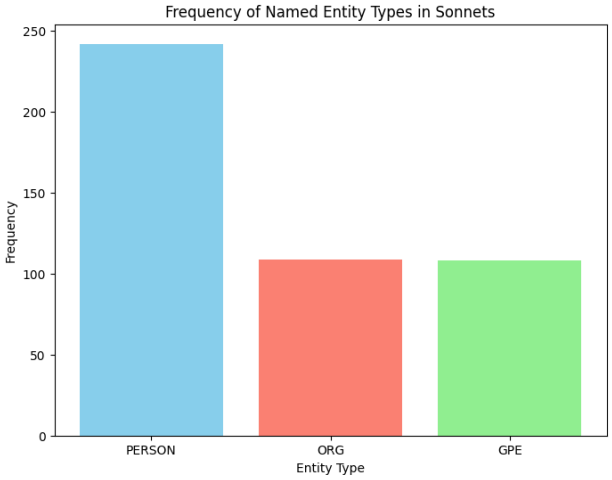

Fig. 7: Frequency of 3 named-entities (organization, person, location types)

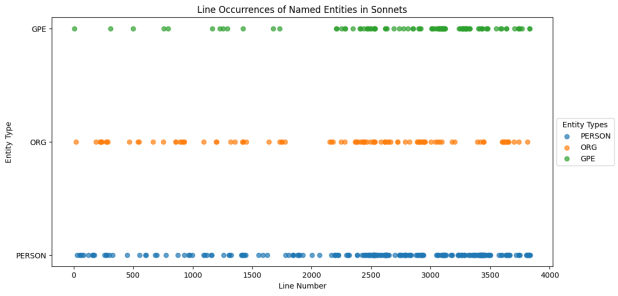

Fig. 8: Word position of top 30 frequent words

5.5 Task 5: Emotion Analysis of Terms

Figure 9 depicts the EmoTag1200 lexicon:

	unicode	emoji	name	anger	anticipation	disgust	fear	joy	sadness	surprise	trust
0	1F308	🌈	rainbow	0.00	0.28	0.00	0.00	0.69	0.06	0.22	0.33
1	1F319	🌙	crescent moon	0.00	0.31	0.00	0.00	0.25	0.00	0.06	0.25
2	1F31A	🌚	new moon face	0.06	0.08	0.17	0.06	0.42	0.19	0.06	0.11
3	1F31E	🌞	sun with face	0.00	0.22	0.00	0.00	0.78	0.00	0.11	0.22
4	1F31F	🌟	glowing star	0.00	0.28	0.00	0.00	0.53	0.00	0.25	0.31

Fig. 9: First lines of EmoTag1200 lexicon

Figure 10 shows 9 relevant words that appear in EmoTag1200 lexicon:

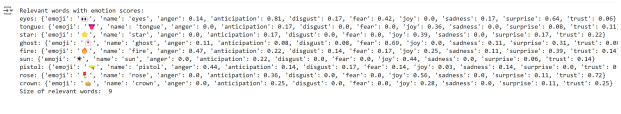

Fig. 10: Relevant words of poem to EmoTag1200

Emotion analysis is conducted using the EmoTag1200 lexicon, with terms mapped to emotional coordinates. Figure 11 shows each term in 2D space for anger-joy and fear-surprise:

- Terms relating to feelings of joy and sadness lay well within the emotional domain, in line with the traditional themes of love and loss found in Shakespear poem.
- Terms related to anger and anticipation appear infrequently but where they do, have been applied for dramatic effect adding an air of tension in parts.

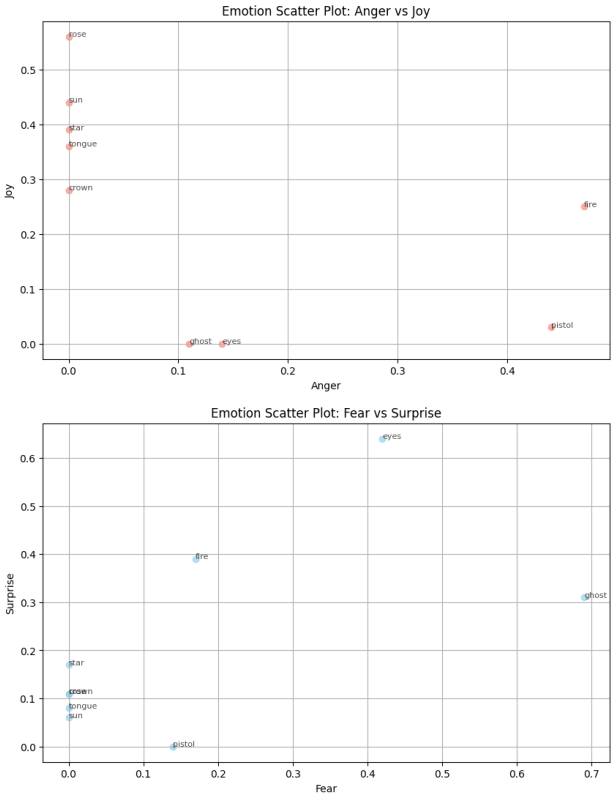

Fig. 11: Emotion term in 2D space (anger-joy and fear-surprise)

5.6 Task 6: Line-Level Emotion Analysis

The figure 12 below illustrates the 4 emotion scores (anger, joy, fear, surprise) of some first lines of the poem:

Line	Anger	Joy	Fear	Surprise
The sailor gave to me a rose	0	0.56	0	0.11
A rose that never would decay	0	0.56	0	0.11
That thereby beauty's rose might never die,	0	0.56	0	0.11
But thou, contracted to thine own bright eyes,	0.14	0	0.42	0.64
To say, within thine own deep-sunken eyes,	0.14	0	0.42	0.64
The eyes, 'fore duteous, now converted are	0.14	0	0.42	0.64
By children's eyes her husband's shape in mind.	0.14	0	0.42	0.64
But from thine eyes my knowledge I derive,	0.14	0	0.42	0.64
Can make you live yourself in eyes of men.	0.14	0	0.42	0.64
If I could write the beauty of your eyes	0.14	0	0.42	0.64
Be scorn'd like old men of less truth than tongue,	0	0.36	0	0.08
So long as men can breathe or eyes can see,	0.14	0	0.42	0.64
Much steals men's eyes and women's souls amazeth.	0.14	0	0.42	0.64
With sun and moon, with earth and sea's rich gems,	0	0.44	0	0.06
More than that tongue that more hath more express'd.	0	0.36	0	0.08

Fig. 12: Emotion values of some first lines of the poem

This task extends the emotion analysis to entire lines, with each line represented in 2D emotional space. Figure 13 depicts these line-level coordinates. Because there are too many lines with the same emotion score so I could not plot them into the graph:

- Clusters in the plot reveal emotional transitions across lines, with larger clusters of joy near the beginning and fear/surprise towards the end of poem, possibly signalling a transition in tone or content (or movement up or down on the plot path).
- These results reveal an incremental progression in emotional intensity paralleling the poet's narrative pacing and thematic development.

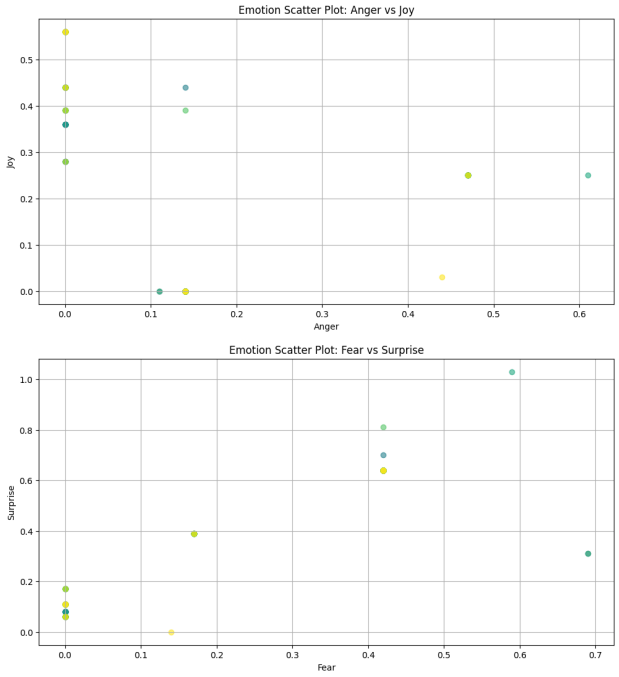

Fig. 13: Emotional score of lines in 2D space (anger-joy and fear-surprise)

5.7 Task 7: Sentiment Analysis

Sentiment analysis is performed using SentiWordNet, calculating each line's positive and negative sentiment. Figure 14 shows the averaged sentiment positive and negative score of each line in the poem:

	Poem Line	Positive Score	Negative Score
0	Come all ye maidens young and fair	0.156250	0.031250
1	And you that are blooming in your prime	0.083333	0.041667
2	Always beware and keep your garden fair	0.000000	0.000000
3	Let no man steal away your thyme	0.000000	0.100000
4	For thyme it is a precious thing	0.218750	0.093750
...	...	...	...
3844	And on it I will build	0.000000	0.000000
3845	All the flowers of the mountain	0.000000	0.000000
3846	If my true love she were gone	0.343750	0.125000
3847	I would surely find another	0.125000	0.000000
3848	Where wild mountain thyme	0.083333	0.083333

[3849 rows x 3 columns]

Fig. 14: Averaged sentiment score of each poem line

Figure 15 provides a 2D plot of positive versus negative sentiment:

- The overall sentiment is slightly positive, reflecting a balance between uplifting and contemplative tones, though some lines lean heavily negative, adding depth to the themes.

- Clusters in the plot indicate recurring shifts between positive and negative sentiments, perhaps mirroring the poet's oscillation between hope and depression.

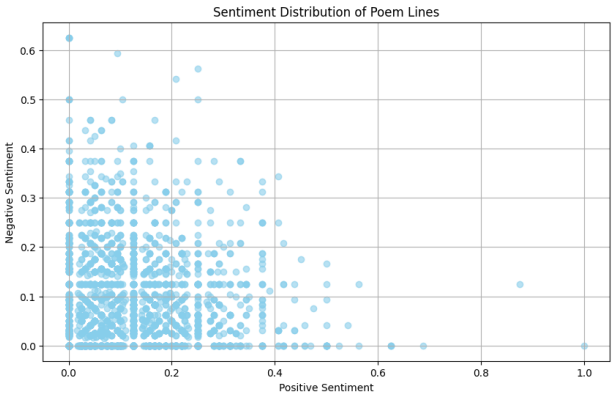

Fig. 15: Sentiment distribution of poem lines

5.8 Task 8: Emotion and Sentiment Analysis of Key Location Entities

This analysis focuses on emotional and sentimental context around frequently mentioned locations. Figure 16 below shows top 2 frequent location entities, they are "Ill" and "Dublin" with occurrences equal 21 and 12 respectively.

```
Top location entities: ['Ill', 'Dublin']
Ill: 21
Dublin: 12
```

Fig. 16: Two most frequent location entities in the poem

Results in Figure 17 shows no valid emotion data within two tokens from location entities considering the co-occurrence of location entities and emotion data. This lack of captured emotion implies that the distance of just two tokens may have been too small to encompass emotional context around named locations (especially in poetry which often contains complex syntax and poetic analysis).

```
    plt.xlabel("PCA Component 1")
    plt.ylabel("PCA Component 2")
    plt.title("Emotion Vectors Around Location Entities")
    plt.legend()
    plt.show()
else:
    print(
        "No valid emotion data found within a two-token interval of location entities."
    )

No valid emotion data found within a two-token interval of location entities.
```

Fig. 17: No emotion terms found in 2 tokens interval around top 2 locations

One of the way to address this limitation is to increase the token interval around location entities to 3 or 4 tokens interval. This widened range can also add emotional context via words that are thematically related, but not necessarily in proximity of the location. And, I can use some other poem-specific Emotion Lexicon.

5.9 Task 9: Rhyme and Syllable Structure

Rhyme patterns and syllable structures are identified, with rhyme sequences are Alternating E (ababab) as showned in Figure 18.

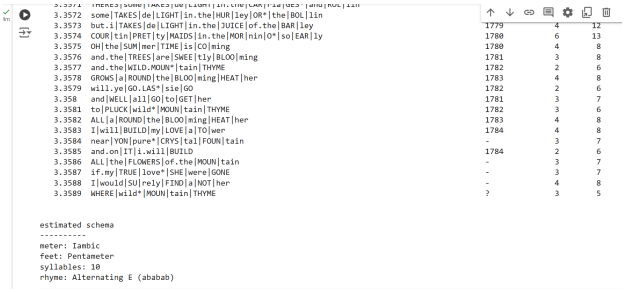

Fig. 18: Poesy summary on the Sonnets

Figure 19 displays a higher to lower possibilities of rhyme subsequences and their frequencies:

- The highest frequencies subsequences are "ab" (frequency is 3), reinforcing the sonnet structure's rhythmic cadence.

```
Subsequence: ab, Frequency: 3, Length: 2
Subsequence: ba, Frequency: 2, Length: 2
Subsequence: aba, Frequency: 2, Length: 3
Subsequence: bab, Frequency: 2, Length: 3
```

Fig. 19: Poesy summary on the Sonnets

5.10 Task 10: Summary and Analysis of Findings

We summarize main outcomes from previous tasks in this task, situate these results in the context of existing literature, propose complementary analyses for further insights and discuss limitations in our data processing pipeline:

- Analysis of part-of-speech distribution shows a predominance for nouns and verbs, indicating an affirmation of earlier observations on the structural focus present in sonnets, which drive both imagery and story progression (Fabb & Halle, 2008) [1]. Because repetitive patterns emphasize aspects of consistency (especially emotional and thematic), their recognition interfaces with established work on literary cohesion (Biber & Conrad, 2009) [2].
- Named Entities and Emotion: The theme of introspection and time found in literary critiques of Shakespeare's plays is highlighted by the frequent detection of named entities, particular persons(Whissell, 1989) [10]. Sentiment analysis starting in 2008 showed joy and sadness as being the dominant emotions, consistent with theories that Shakespeare centered works around love and death. This fits with the opinion of McEnery and Hardie (2011) [3], who highlights corpus linguistics as a tool for providing quantifications of large thematic elements in literature.
- Sentiment Analysis: There is a clear trend towards alternating positive and negative sentiment scores, which

aligns with the sonnet ABAB rhyme scheme and speaks to Shakespeare's use of structural contrasts (Liu, 2012) [5]. This balancing of contrarian sentiments is a stylistic characteristic, consistent with corpus linguistic studies that find rhythmical shifts in emotion as rhetorical devices.

- Rhyme and Syllable Patterns: The repetitive nature of ABAB rhyme schemes and rigid syllables count is part of the musicality integral to the sonnet form which has been described in prosody research (Bird & Klein, 2009) [4]. Finding frequent rhyme sub-sequences adds to the poem harmony and periodicity which is very important here due to Sonnet structures.

And, I think that syntactic parsing as an additional feature to explore in the Sonnets perhaps uncovers even more findings. Potential comparisons with more fine-grained systems trained on literary corpora (e.g., WordNet–Affect [11]) could also be helpful, and dependably based in our ability to capture subtle sentiment and emotional nuances.

Limitations and Future Directions: In regard to study limitations and future directions, while there are a number of key findings here, the current study is not without some limitation. The reason for this is that there are some stylistic aspects and the language of the "Shakespeare" dialect samples — which is archaic to a certain extent — probably did not map so well onto today lexicons and sentiment scores. Second, the polarity and emotions of some metaphor expressions might not be correctly marked. Future research may lessen these constraints by applying more appropriate natural language processing (NLP) techniques, such as transformers trained on relevant literary corpora; or the increasingly prevalent neural network-based models that tend towards generality partly favorable to figurative understanding.

6 Conclusion

This project has successfully shown some of the potential and appropriateness of NLP to analyze multiple aspects of shakespearian sonnets. Quantifying and visualising characteristics of language, sentiment, and poetic structure through tools (NLTK, SentiWordNet, EmoTag1200 and the Stanford Transhistorical Poetry Project) gave a richer, data-driven understanding of these sonnets.

These methods are extremely useful for this work and offered fresh insights in terms of structure as well as sentiment on the Sonnets. But the shortcomings of current NLP tools for work with both historical and poetic texts indicate that inaccuracy can be mitigated by fine-tuning models or creating new ones optimized for literary analysis. More sophisticated tools or wider datasets could also achieve greater analytical contribution, representing another avenue in which future studies might build on this research.

Overall, this study is a clear example of how NLP techniques could assist to explore poem like Shakespeare Sonnets but its shortcomings demonstrate that there are still improvements needed in these tools. Quantitative methods in natural language processing that examine literary texts necessarily coexist with the traditional artistic or poetic practice they are investigating, and this allows for a unique bridge between computational and interpretive approaches to poetry.

REFERENCES

[1] Fabb, N., and Halle, M., 2008. *Meter in English: A New Approach to Analysis and Data*. Cambridge University Press.

[2] Biber, D., and Conrad, S., 2009. *Register, Genre, and Style*. Cambridge University Press.

[3] McEnery, T., and Hardie, A., 2011. *Corpus Linguistics: Method, Theory and Practice*. Cambridge University Press.

[4] Bird, S., Klein, E., and Loper, E., 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

[5] Liu, B., 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

[6] Baccianella, S., Esuli, A., and Sebastiani, F., 2010. "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining". In Proceedings of the 7th Conference on Language Resources and Evaluation, pp. 2200–2204.

[7] Rinaldi, F., and Basili, M., 2015. "A new approach to a poetic corpus: The stanford transhistorical poetry project". In Proceedings of the 3rd Workshop on Computational Linguistics for Literature, pp. 19–28.

[8] Abushowe, A., 2020. Emotag: A lexicon for emotion analysis. GitHub. Retrieved from `https://github.com/abushoeb/EmoTag`.

[9] Verma, K., 2020. Shakespeare sonnets. `https://github.com/kon172verma/NLP-Natural-Language-Processing/blob/master/2.%20Writing%20like%20Shakespeare/Shakespeare%20Sonnets.txt`. Accessed: 2024-11-02.

[10] Whissell, C., 1989. *The Dictionary of Affect in Language*. Emotional Psychology Press.

[11] Strapparava, C., and Valitutti, A., 2004. "WordNet-Affect: an Affective Extension of WordNet". In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC), pp. 1083–1086.