

Harry Potter

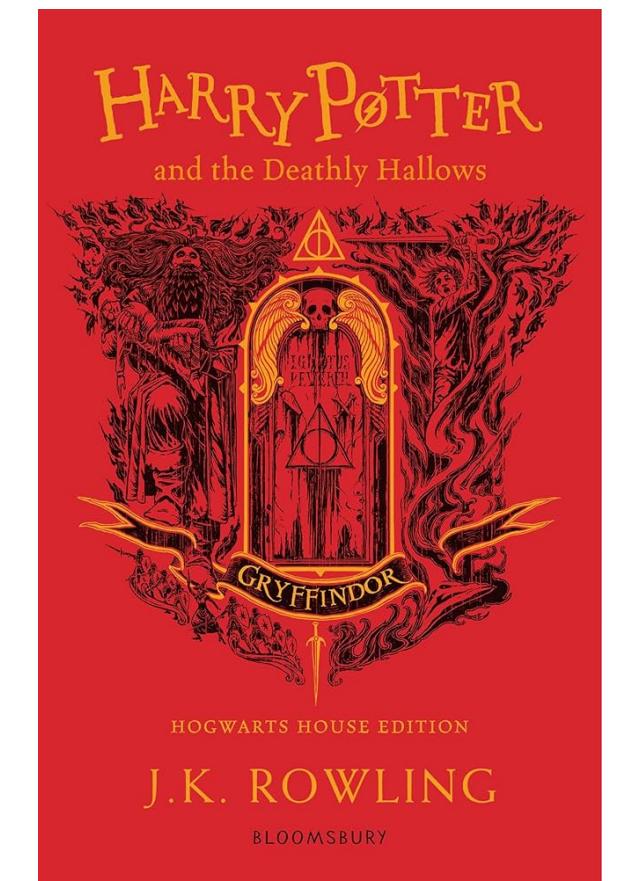
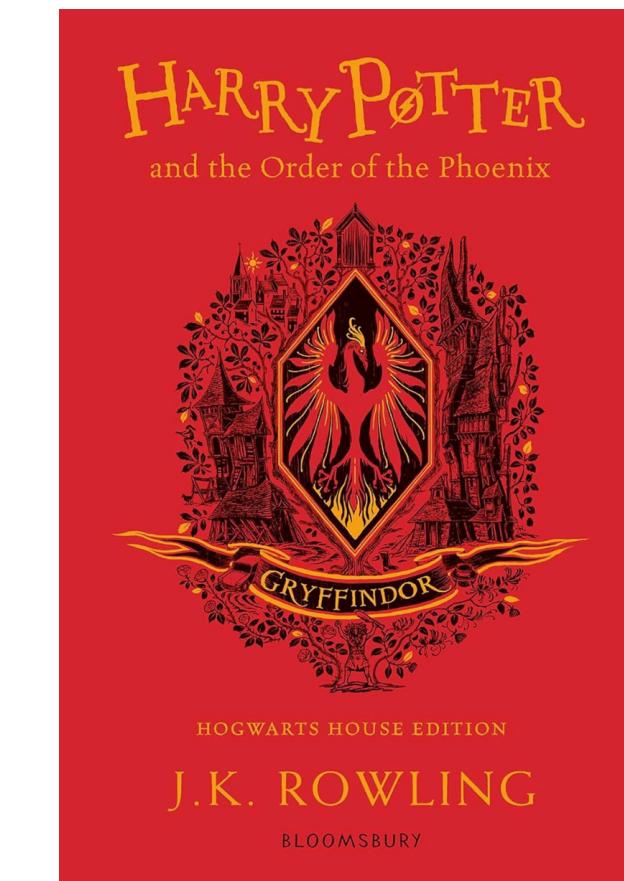
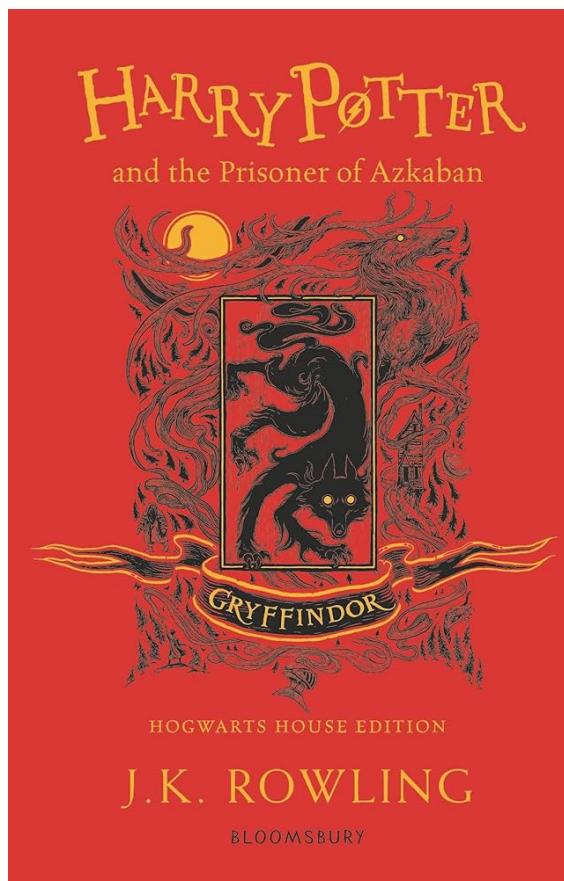
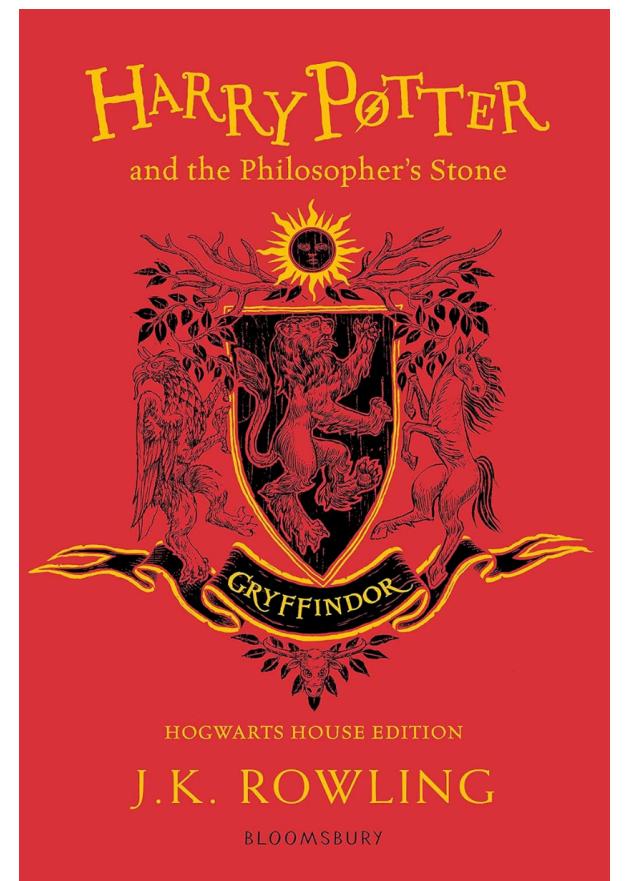
and the Text mining project

Enrico Carraro

Alex Cecchetto

Virginia Murru

1. The Data



1998

2000

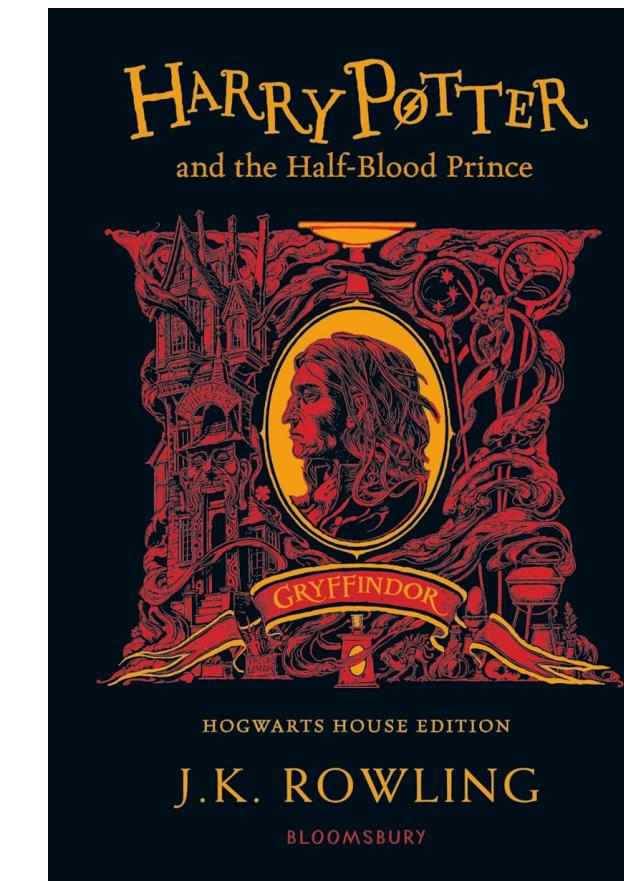
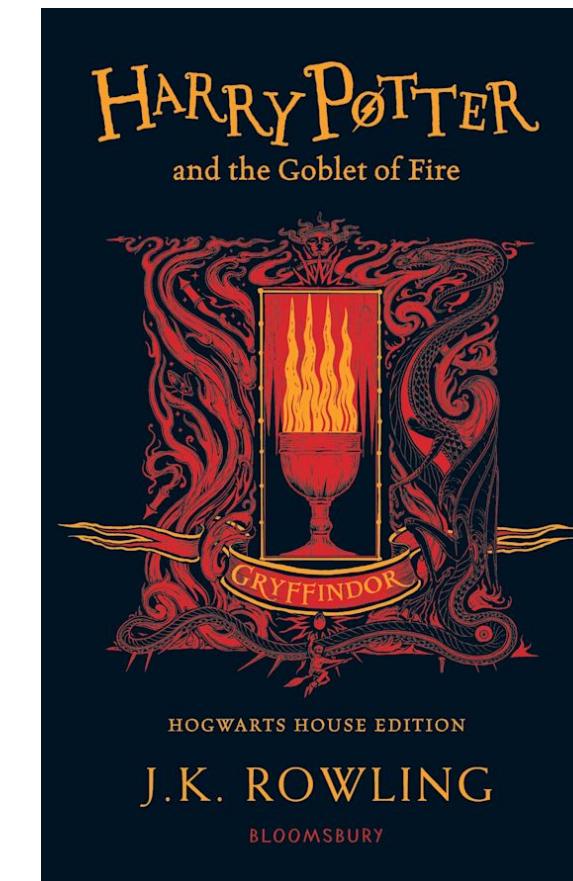
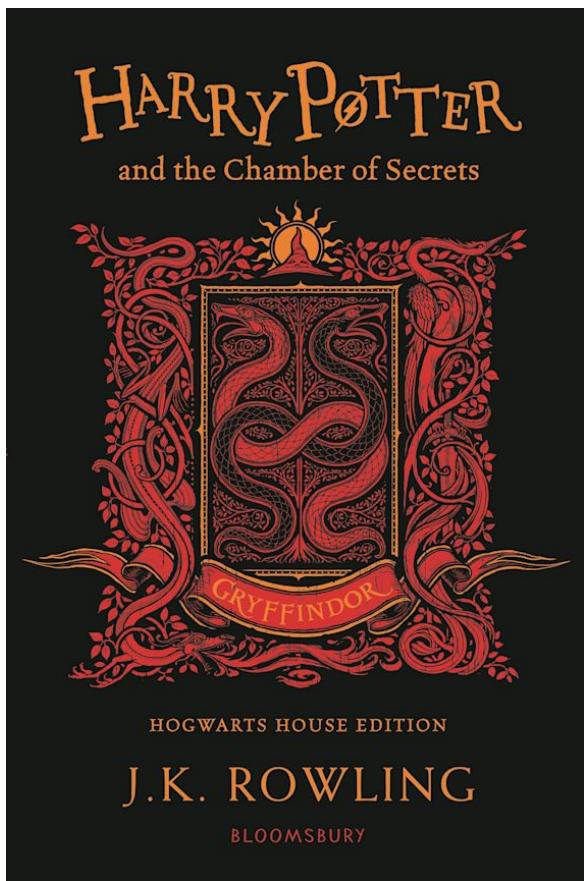
2005

1997

1999

2003

2007

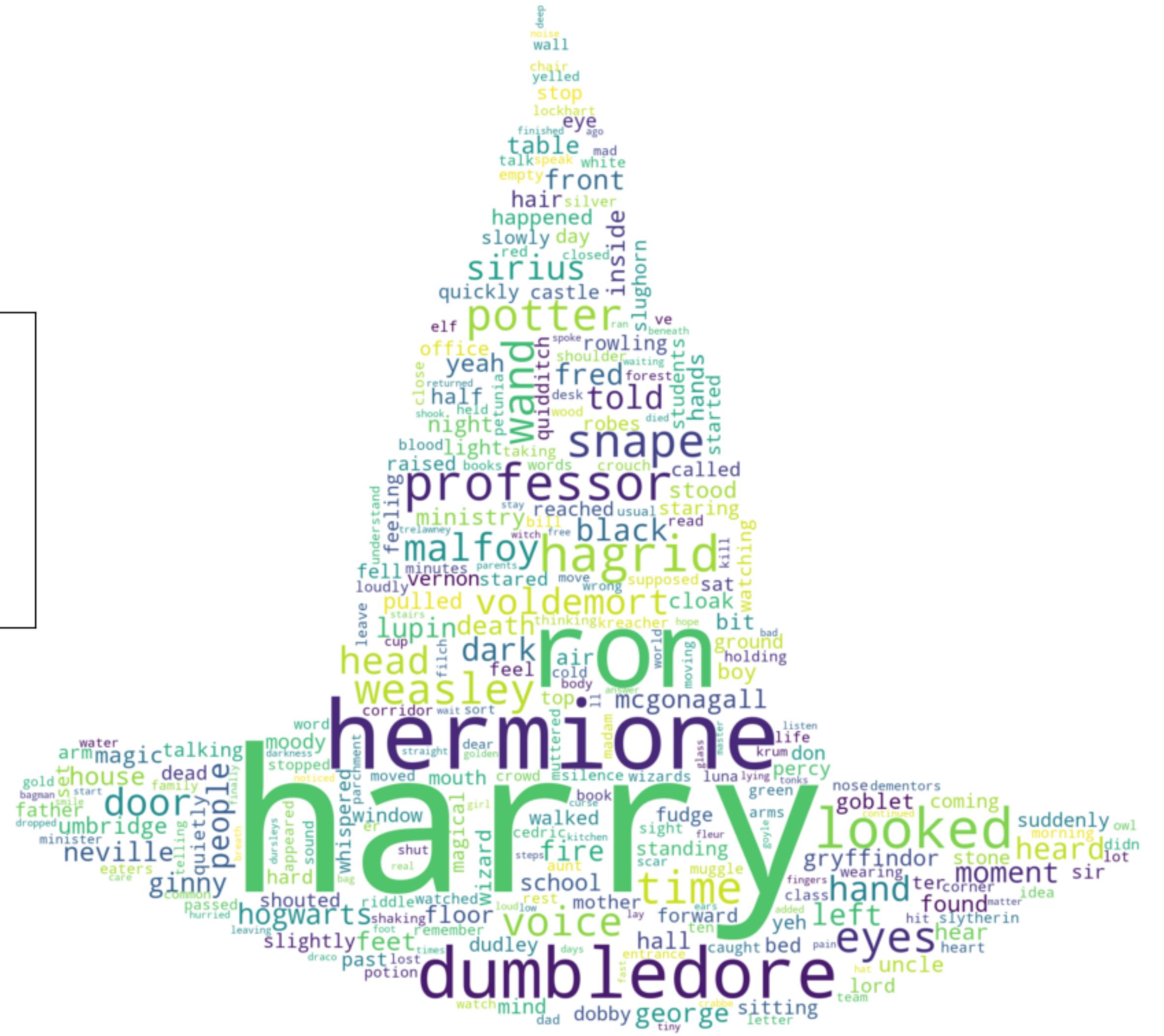
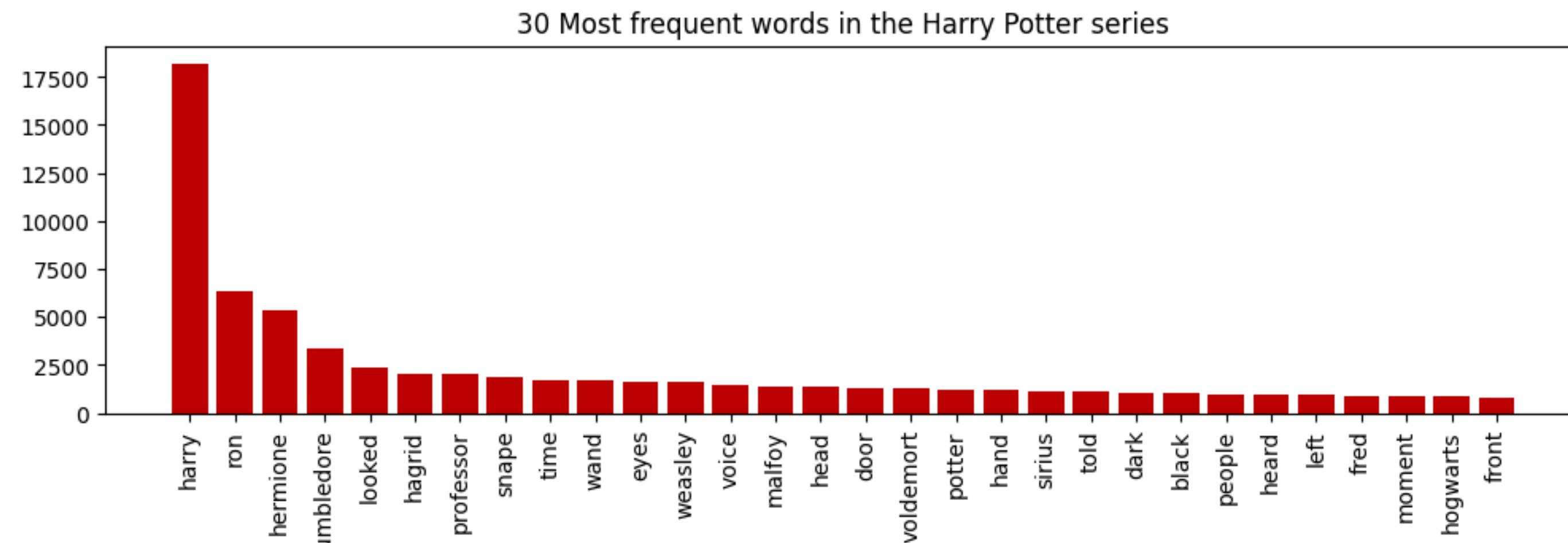


2. Preprocessing

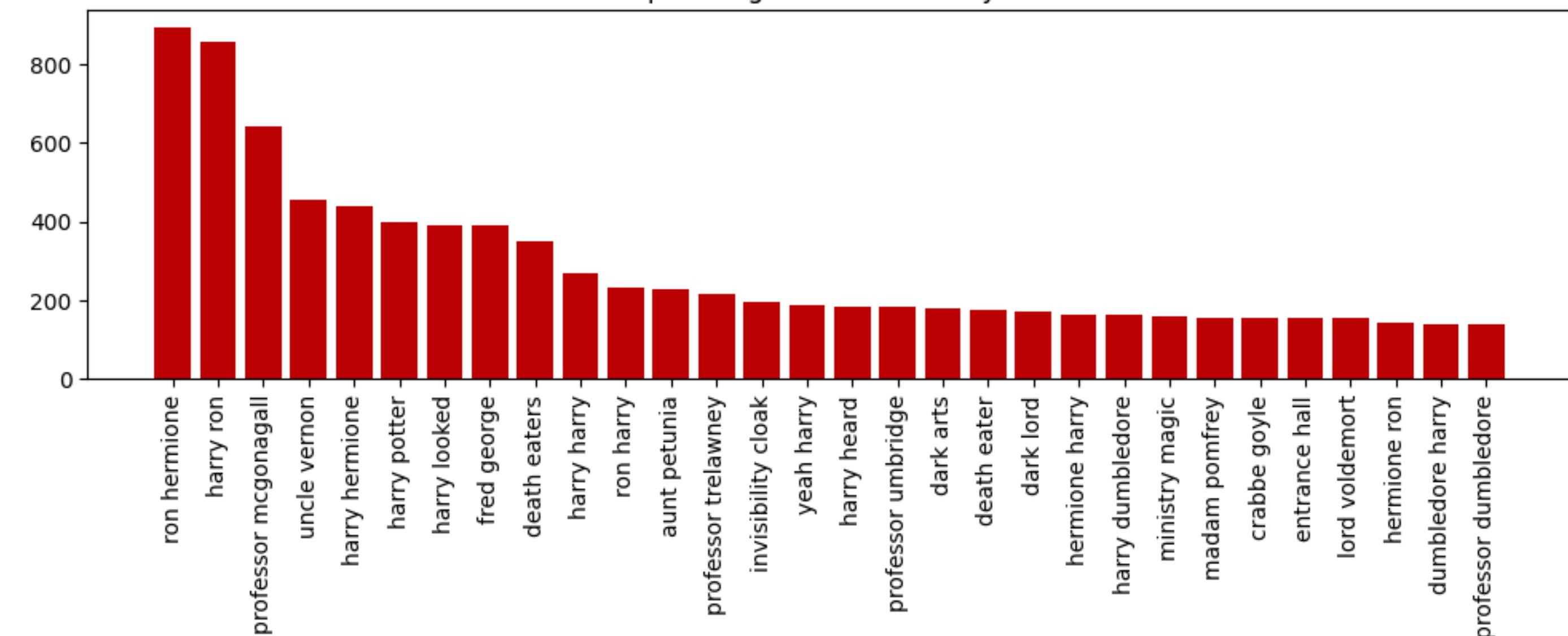
- Obtain the tokens from the books;
- Remove stop-words and special characters;
- Obtain the stemmed words

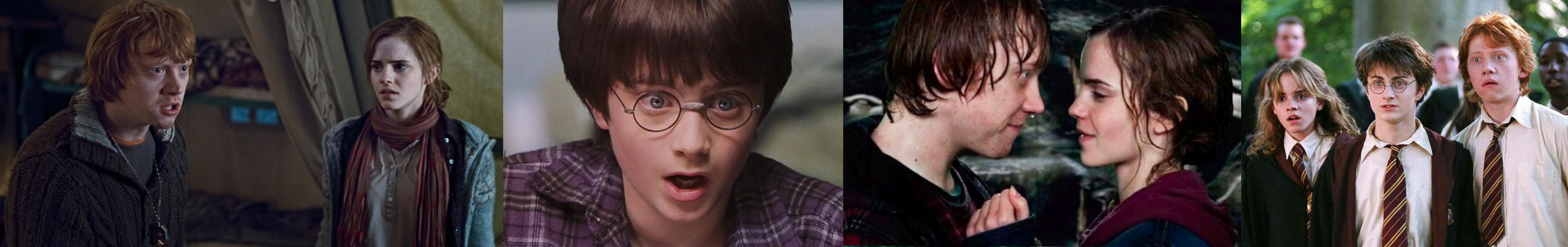
```
class CustomTokenizer:  
    def __init__(self):  
        self.patterns = [  
            (r'[.,;!?]', 'PUNCTUATION'), # Matches common punctuation  
            (r"\b\w+'t\b|\b\w+\b|'\w+\b", "WORD") #specific pattern for words  
        ]  
  
    def tokenize(self, text):  
        tokens = []  
        for pattern, token_type in self.patterns:  
            regex = re.compile(pattern)  
            matches = regex.finditer(text)  
            for match in matches:  
                tokens.append((match.group(), token_type))  
        return tokens
```

3. Most common words



3. Bigrams

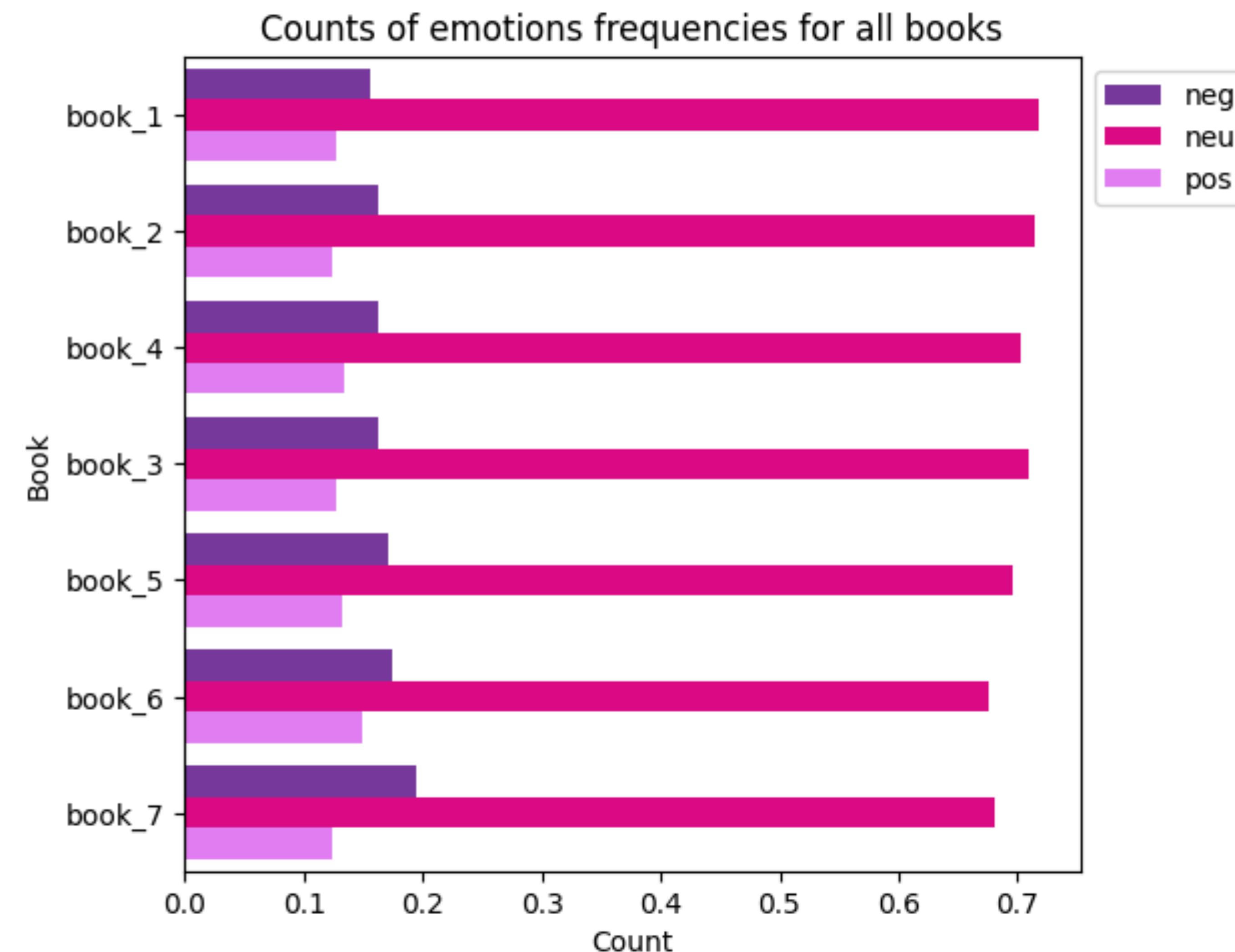




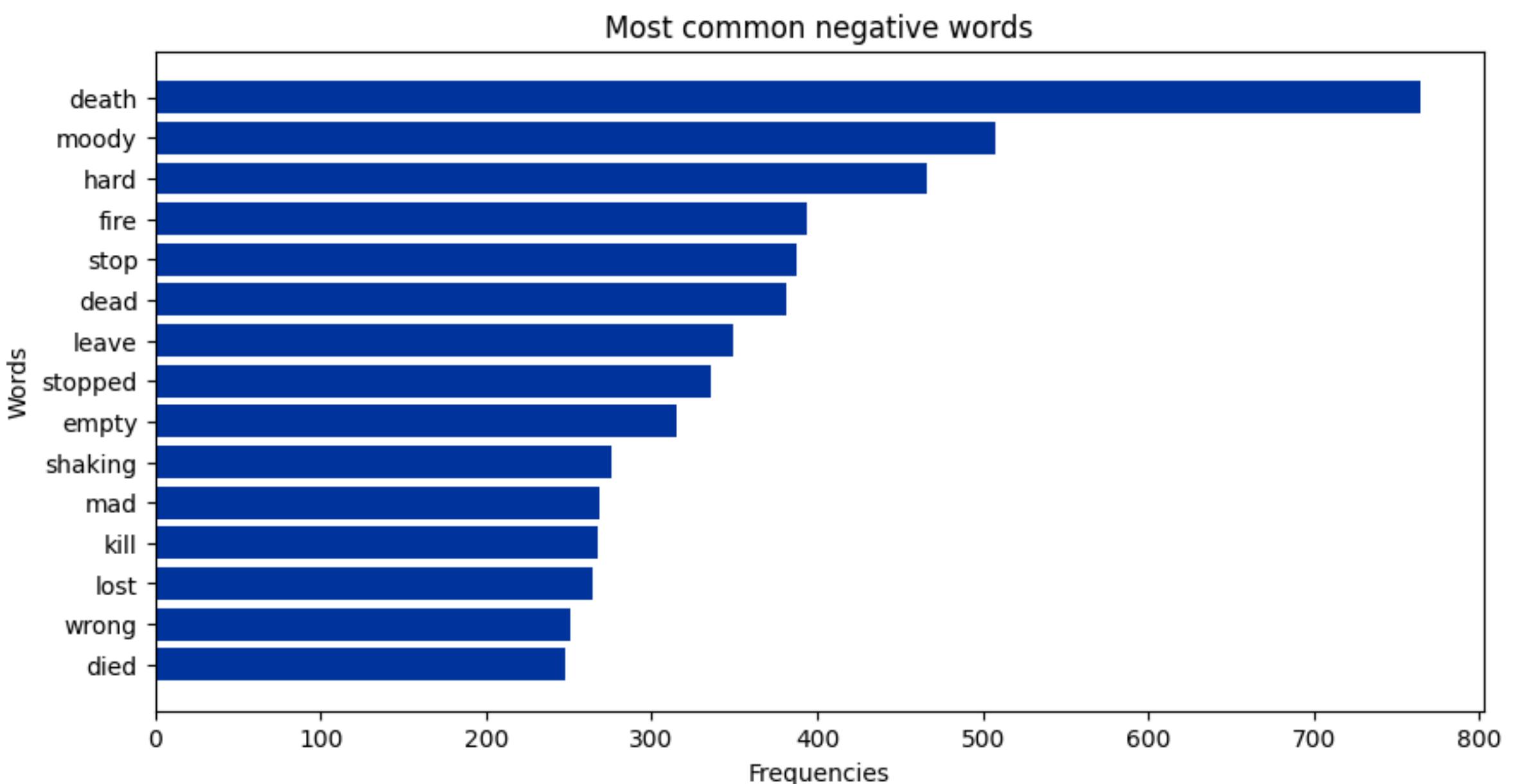
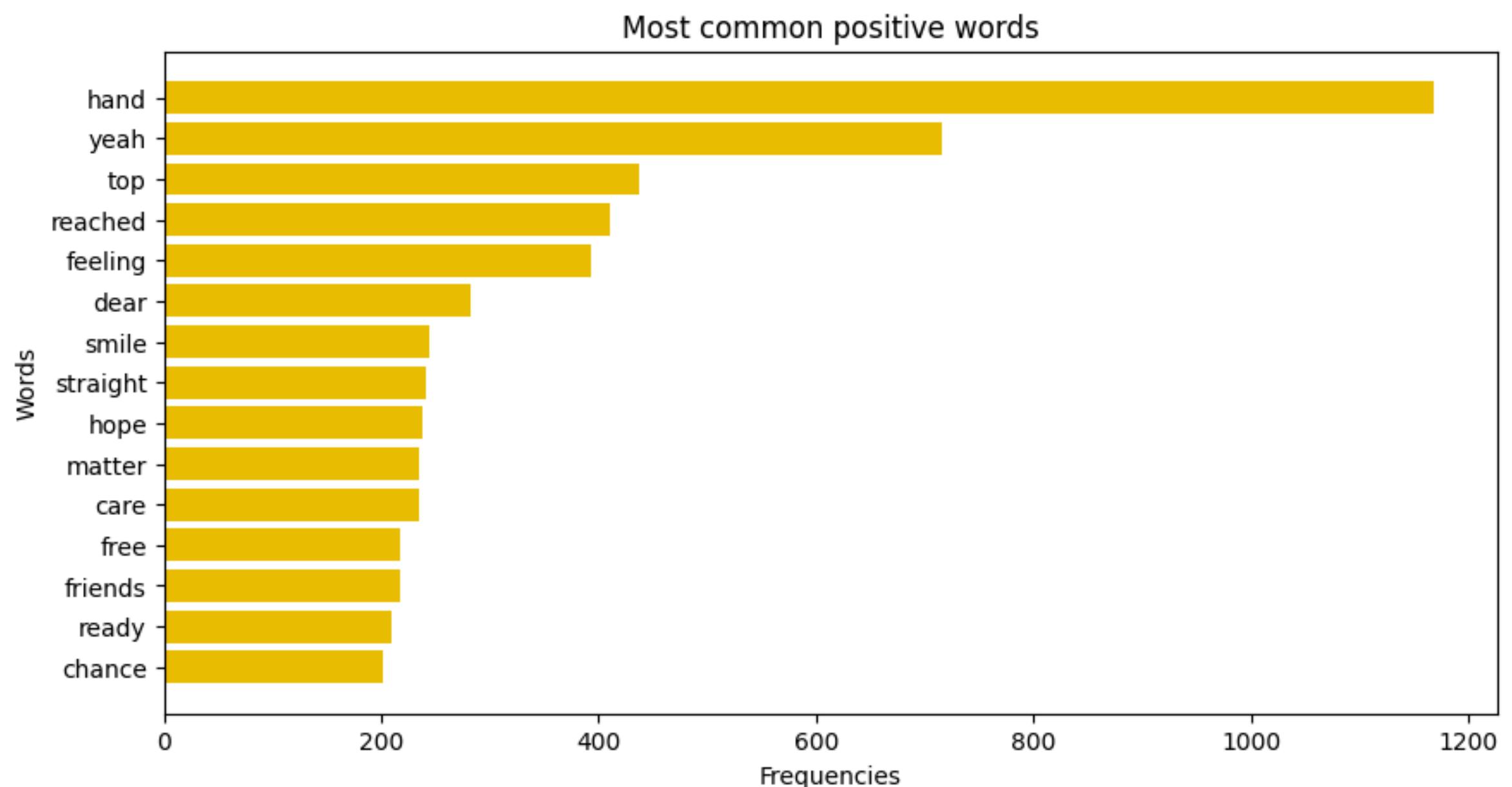
SENTIMENT ANALYSIS



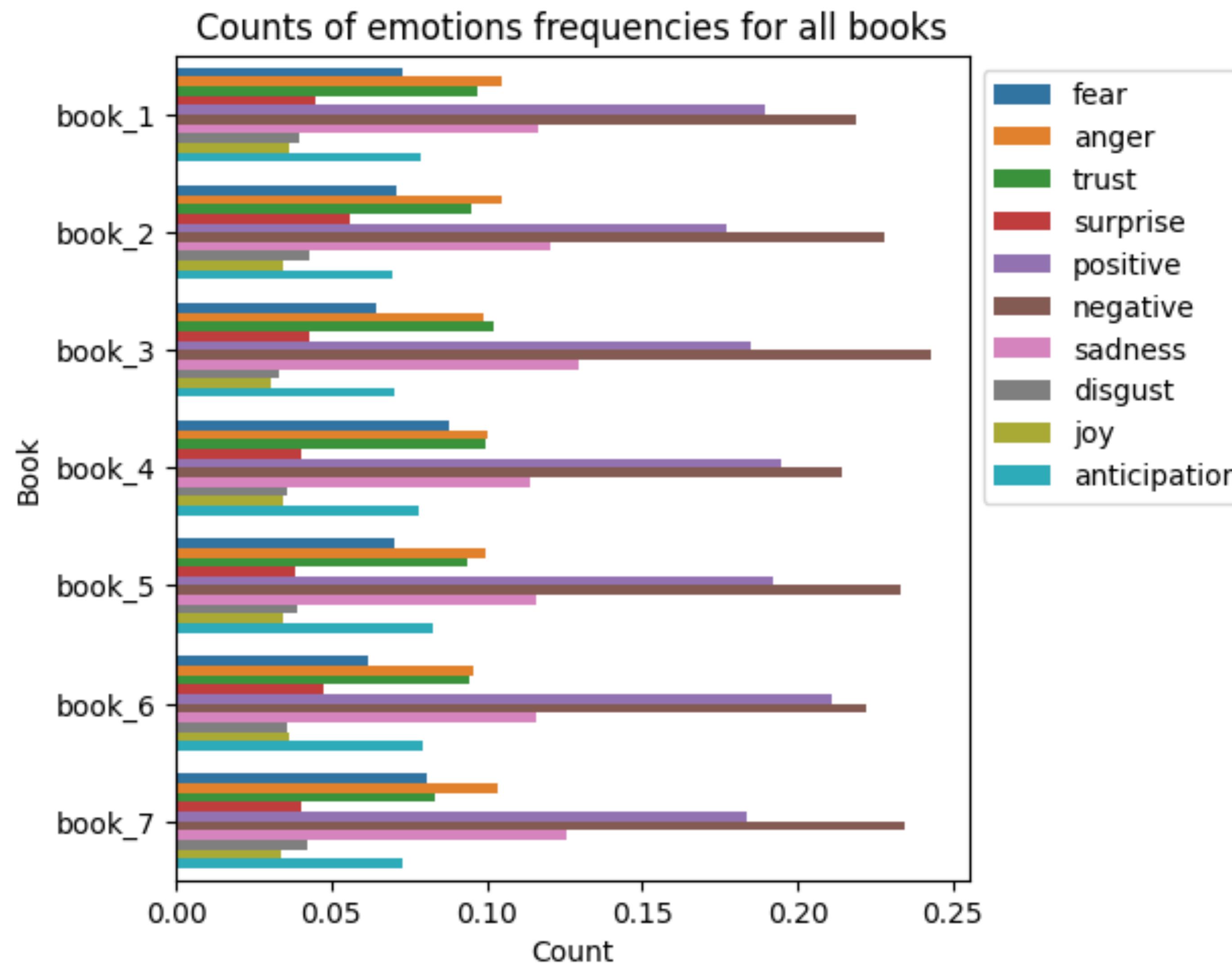
3. Sentiment Analysis



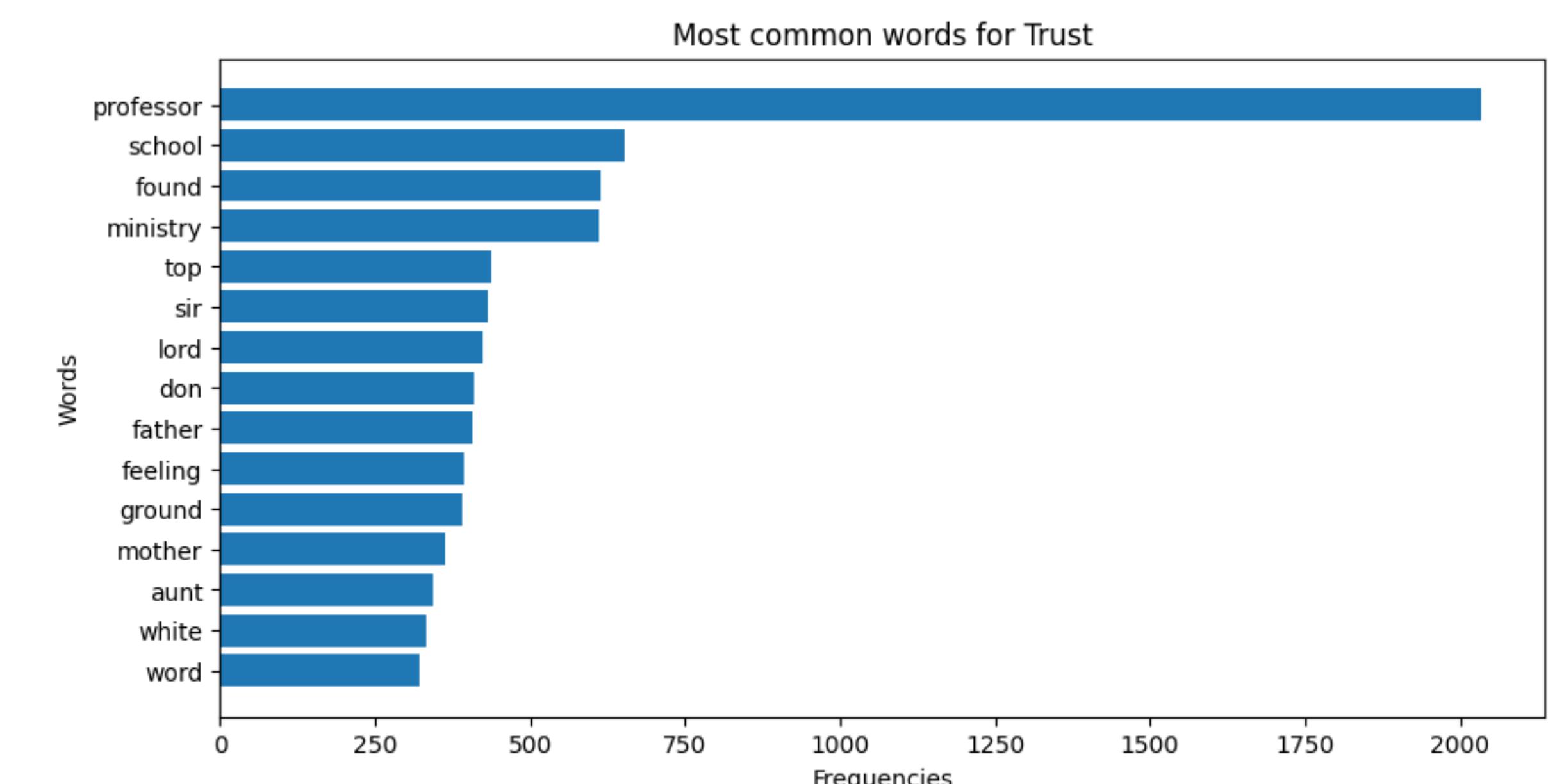
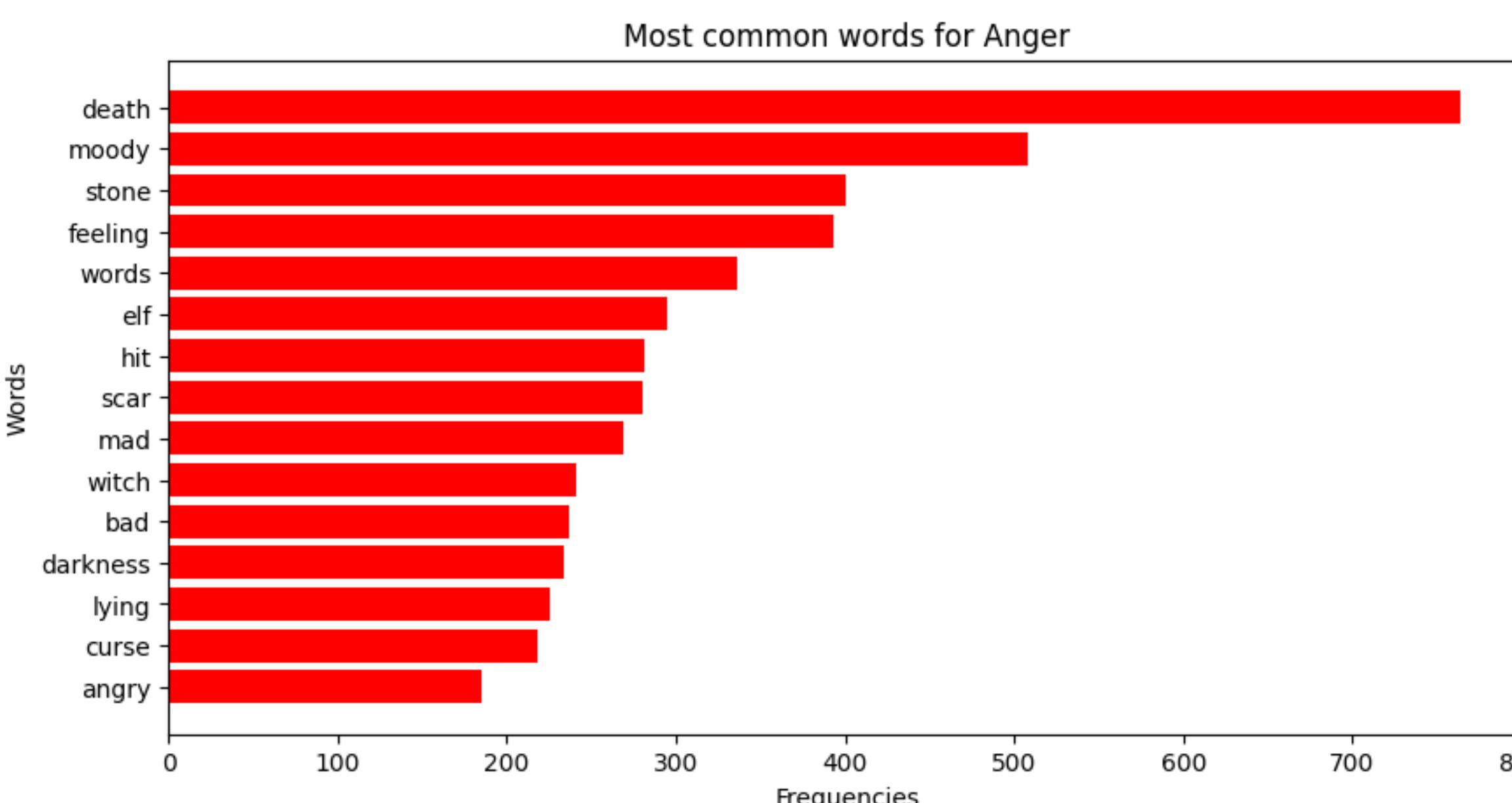
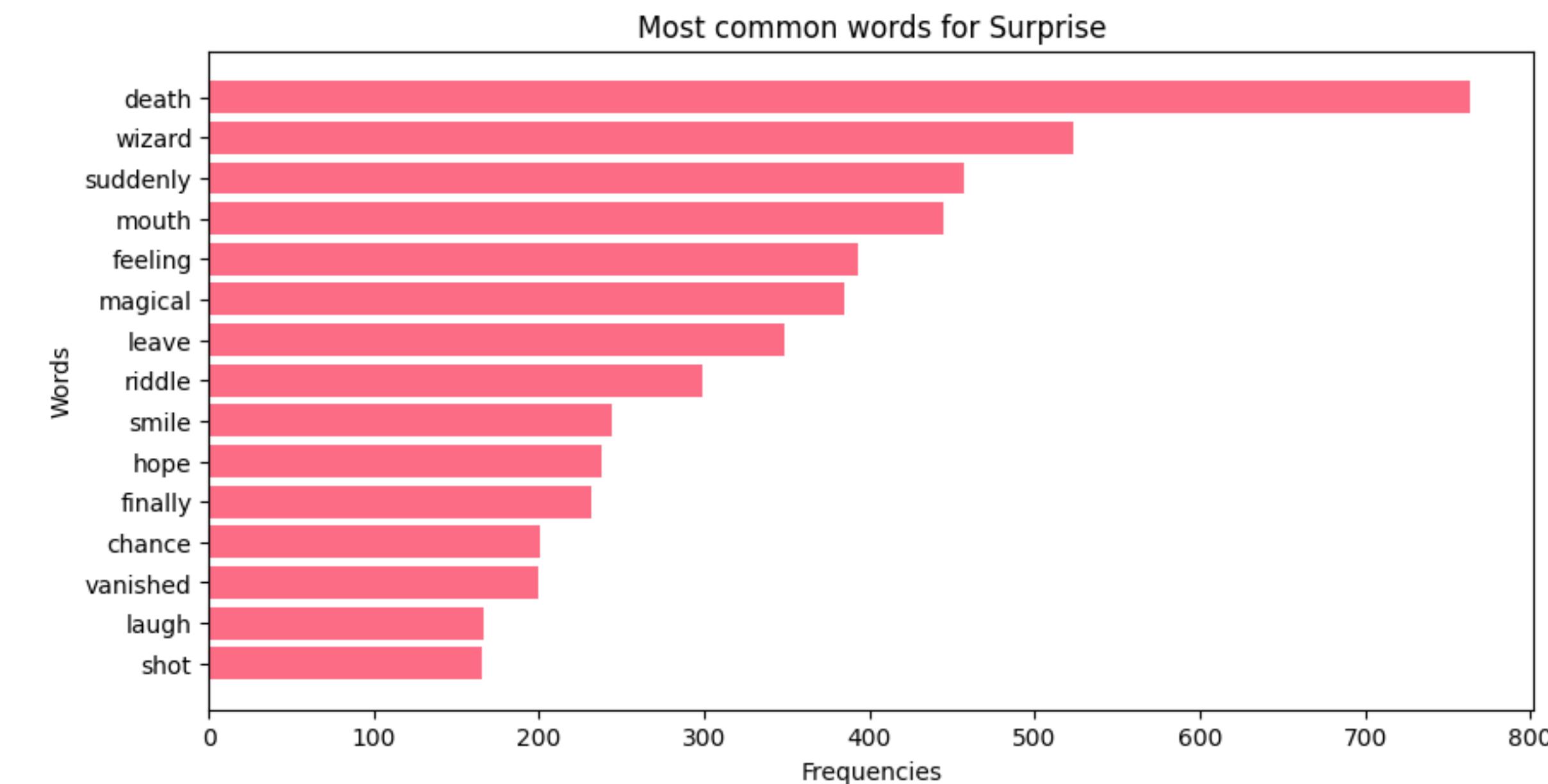
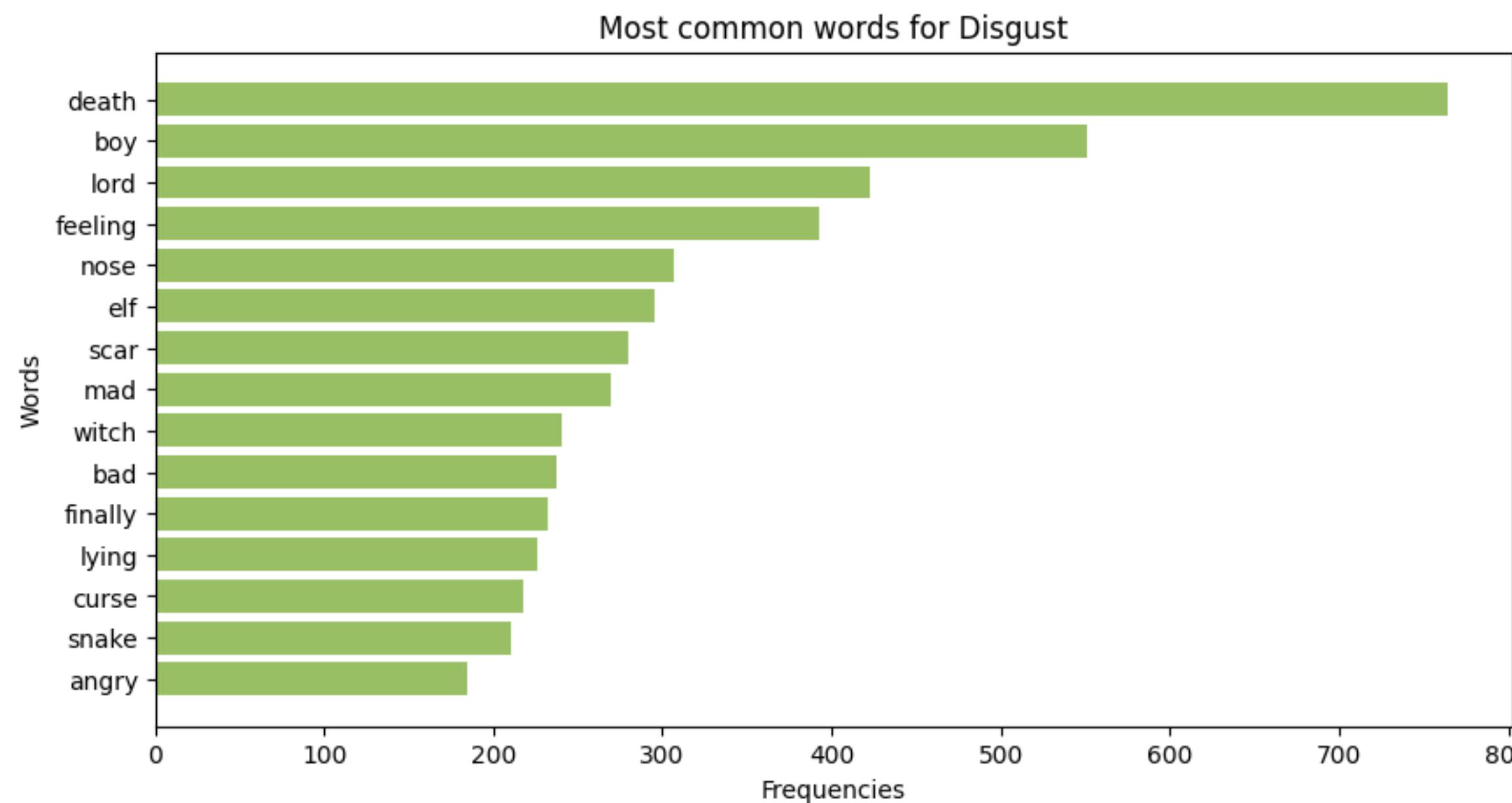
3. Sentiment Analysis



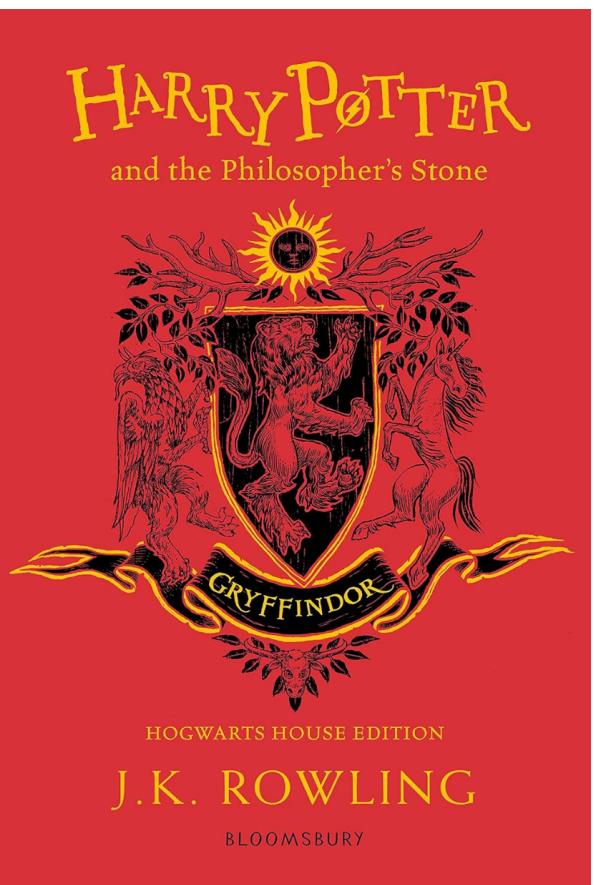
4. Sentiment analysis



3. Sentiment Analysis



4. Books order



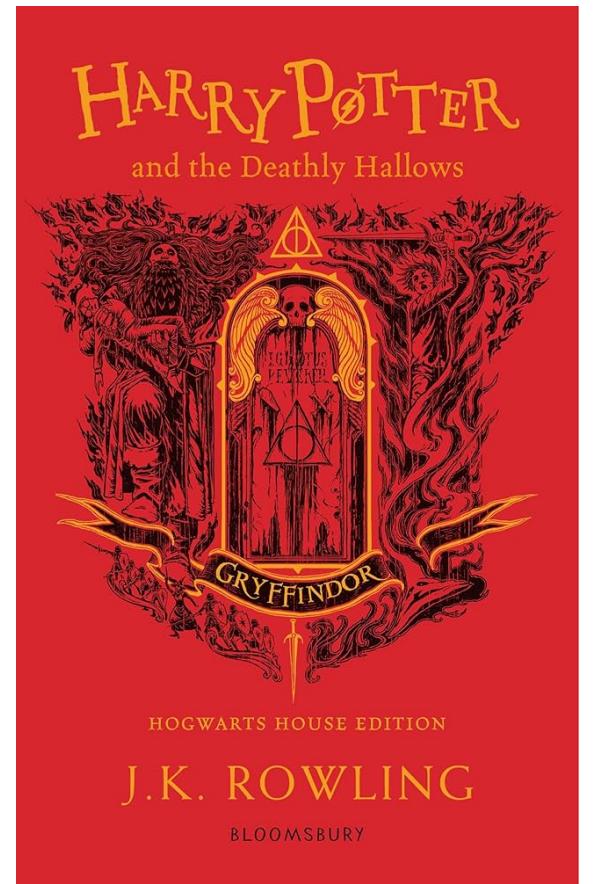
?

?

?

?

?



Harry Potter
and the philosopher stone

Harry Potter
and the deathly Hallows

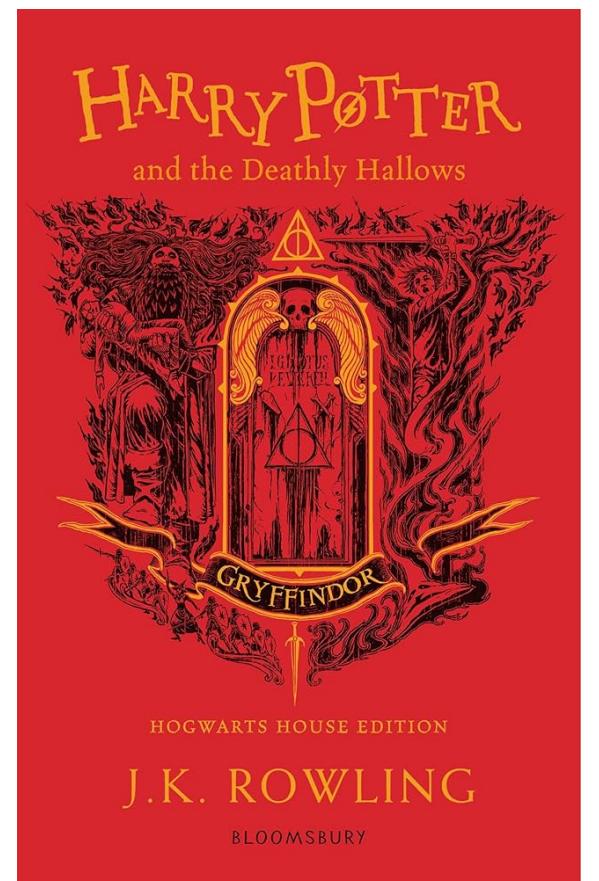
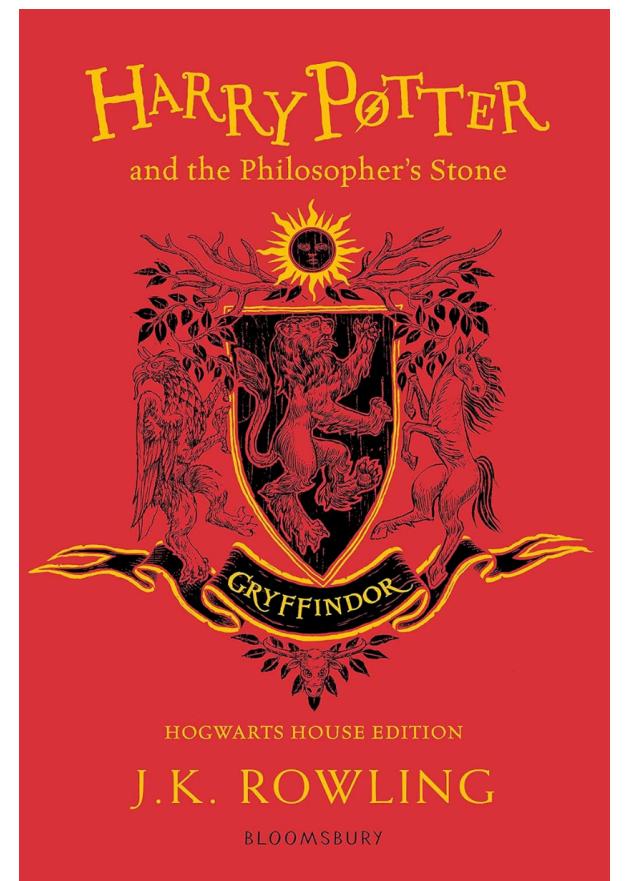
4. Books order - The algorithm

→ Populations: W_2, W_3, W_4, W_5, W_6 ;

→ Scores: $\bar{s}_k = \frac{1}{N_k} \sum_{i=1}^N n_i \log \frac{\theta_{7i}}{\theta_{1i}} = \sum_{i=1}^N n_i s_i$

→ Variances: $\hat{V}(\bar{s}_k) = \frac{1}{N_k(N_k - 1)} \left(\sum_{i=1}^N n_i s_i^2 - \frac{1}{N_k} \left(\sum_{i=1}^n n_i s_i \right)^2 \right)$

4. Books order – Results



1999

1998

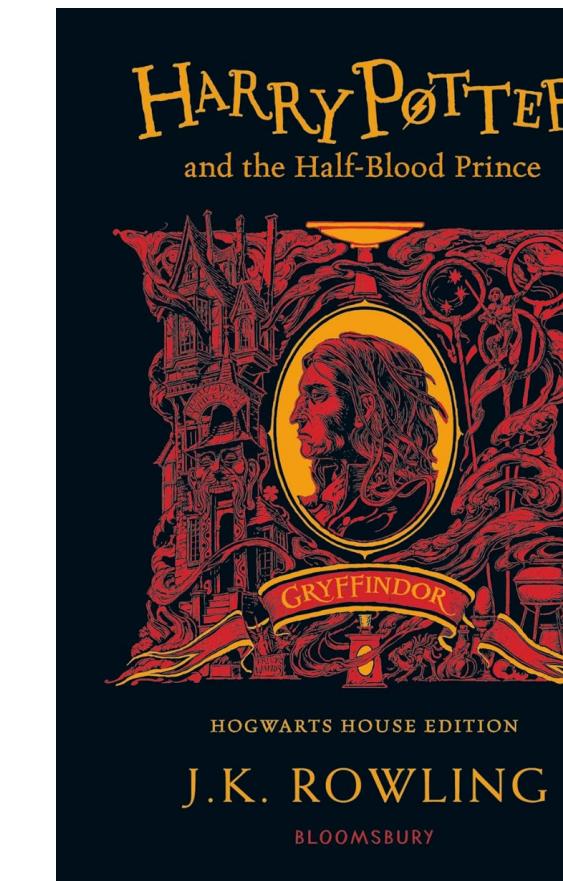
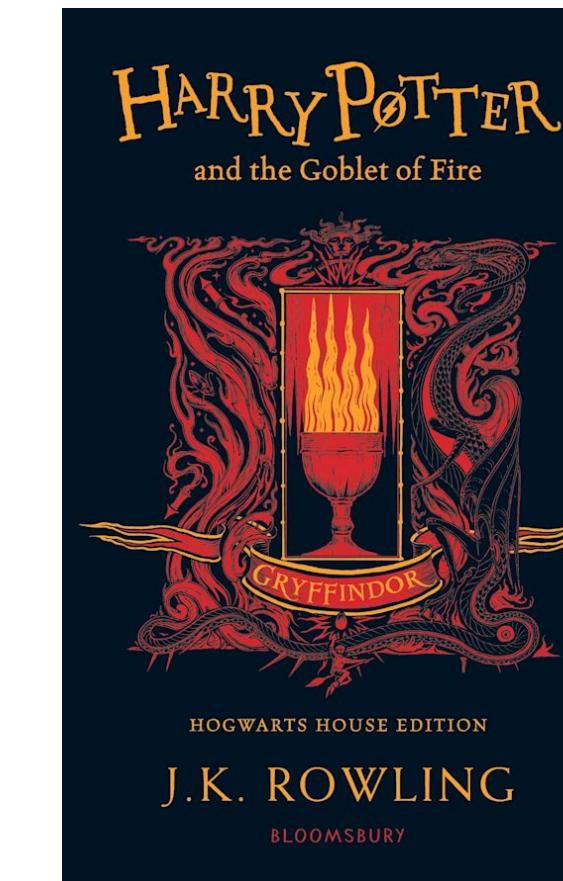
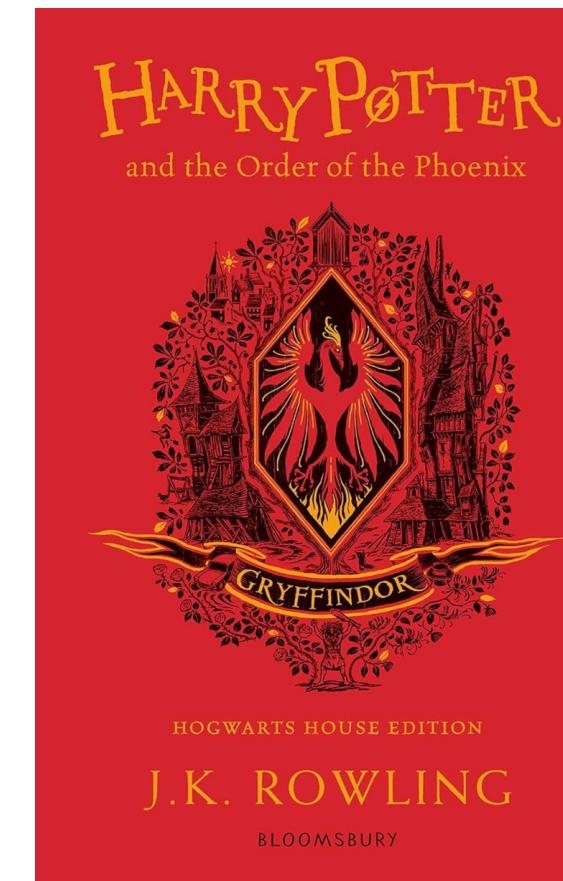
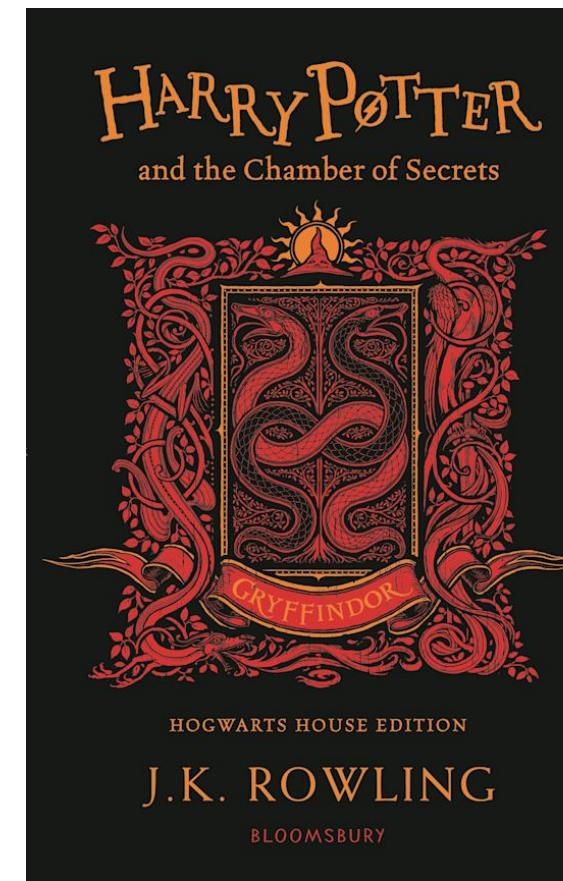
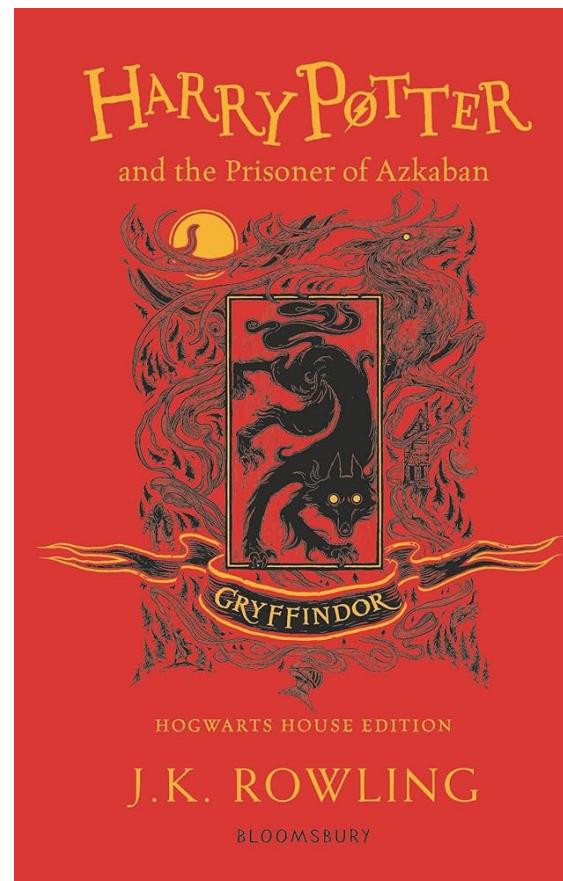
2003

2000

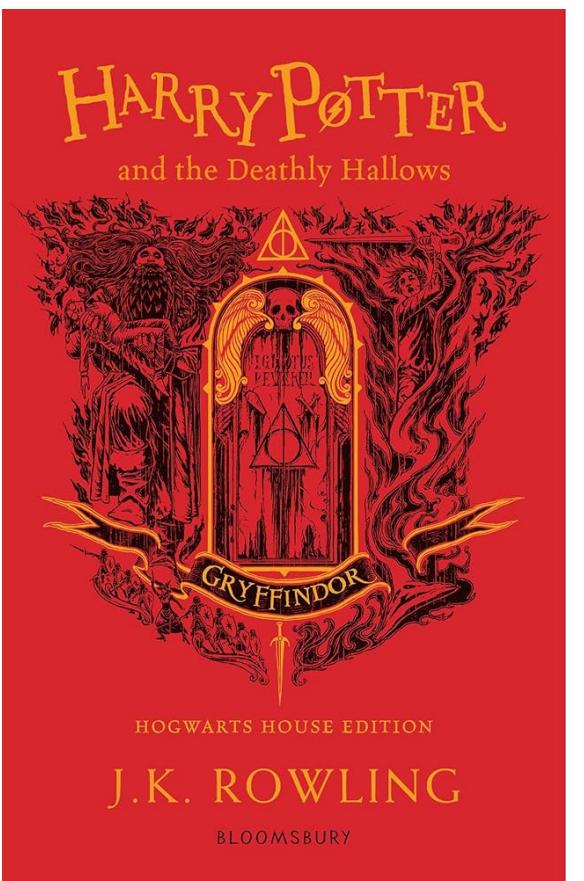
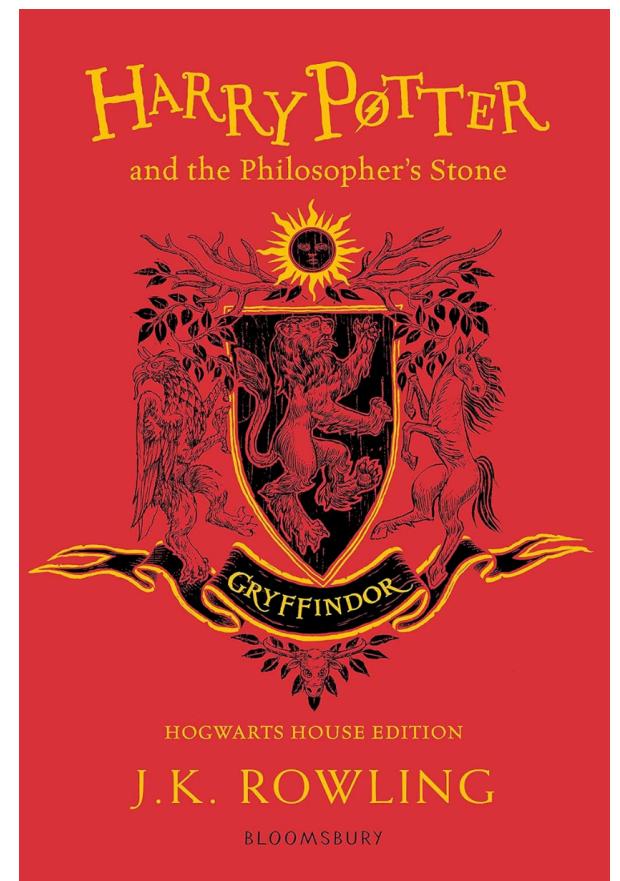
2005

2007

1997



4. Harry Potter and the order of the books



1999

1998

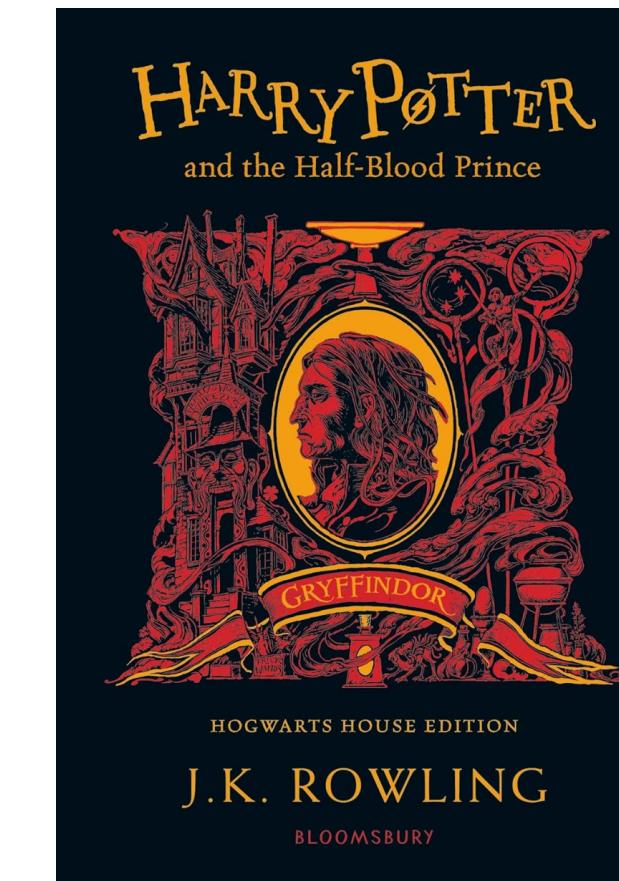
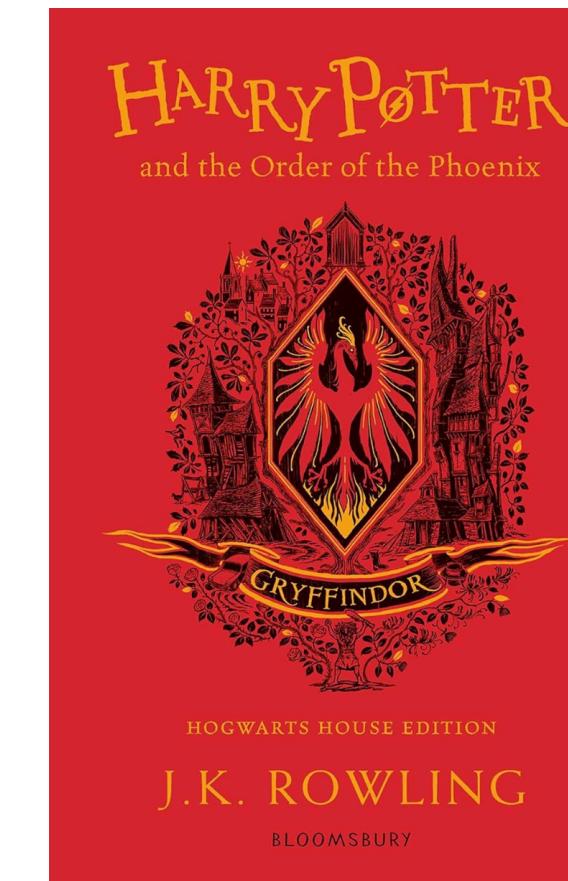
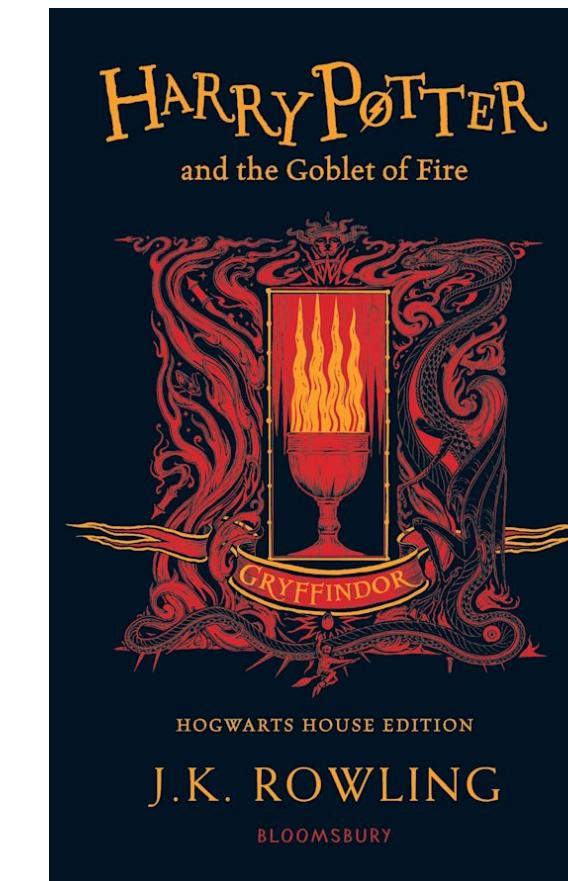
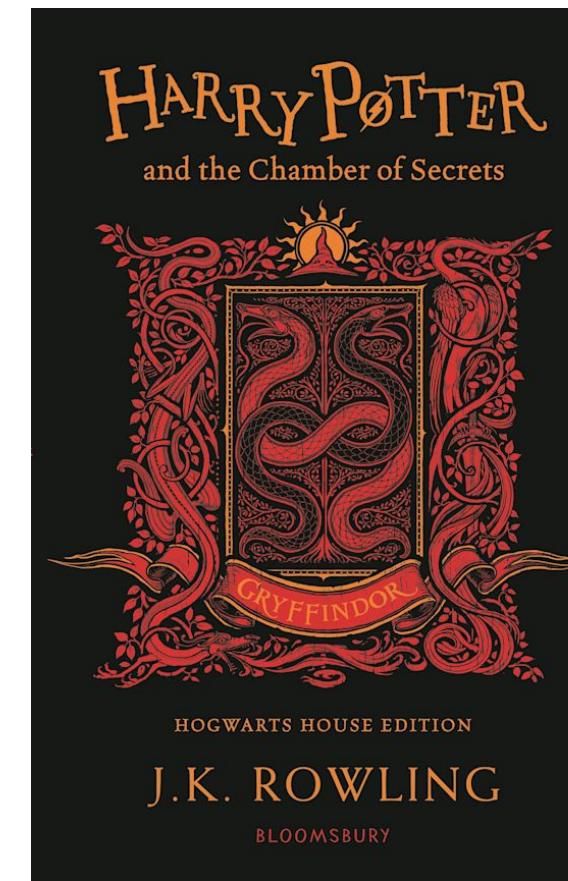
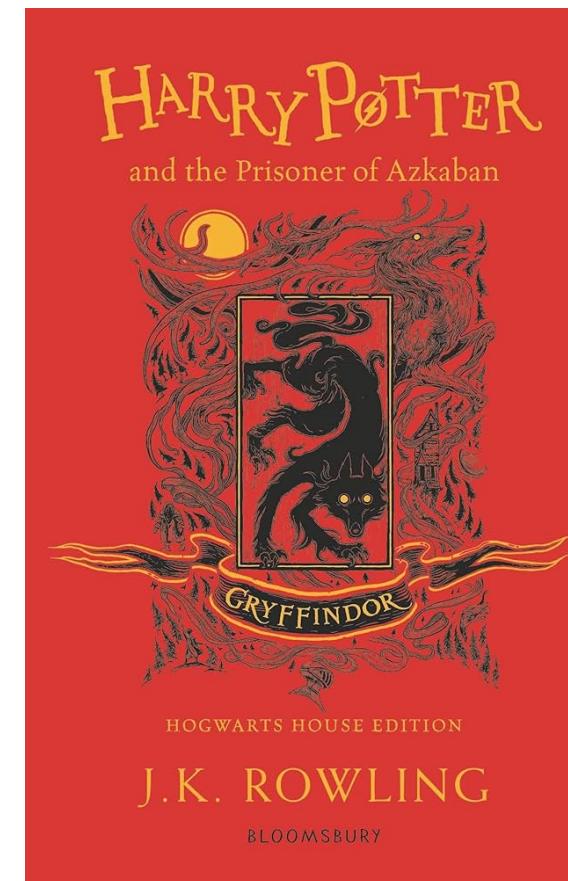
2000

2003

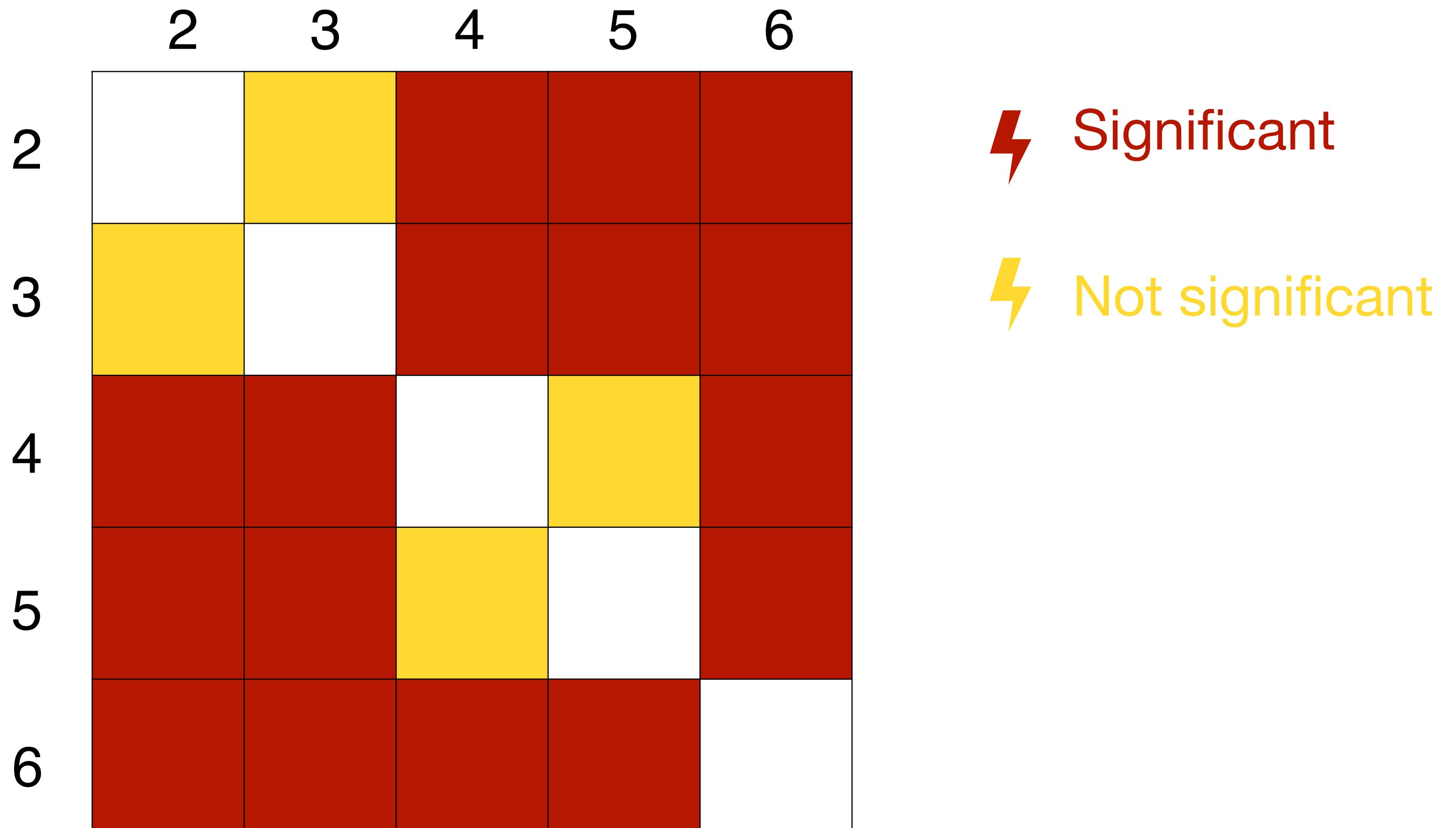
2005

2007

1997



4. Harry Potter and the order of the books



5. Clustering

The similarities between books has been evaluated using the cosine similarity:

$$d_{cos}(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^p x_i y_i}{\sqrt{\sum_{i=1}^p x_i^2} \sqrt{\sum_{i=1}^p y_i^2}}$$

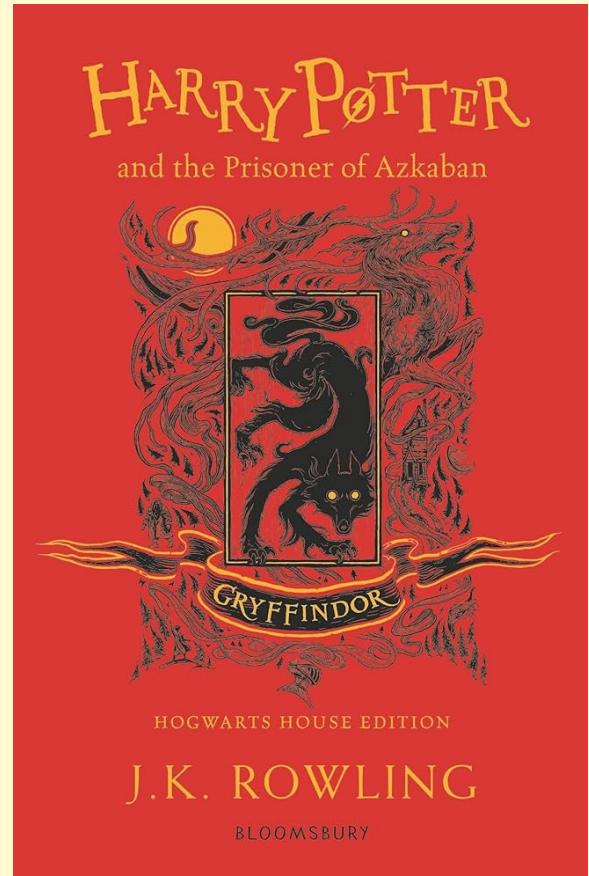
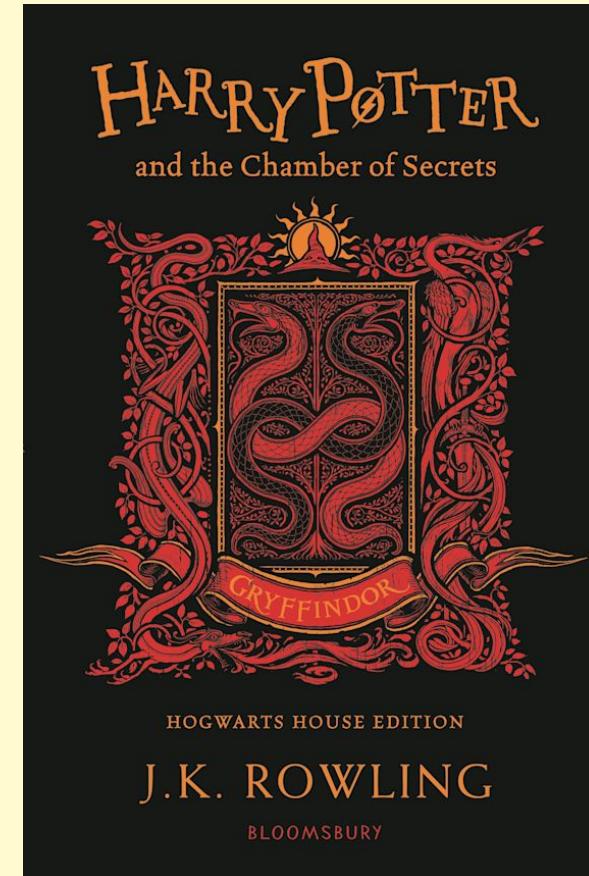
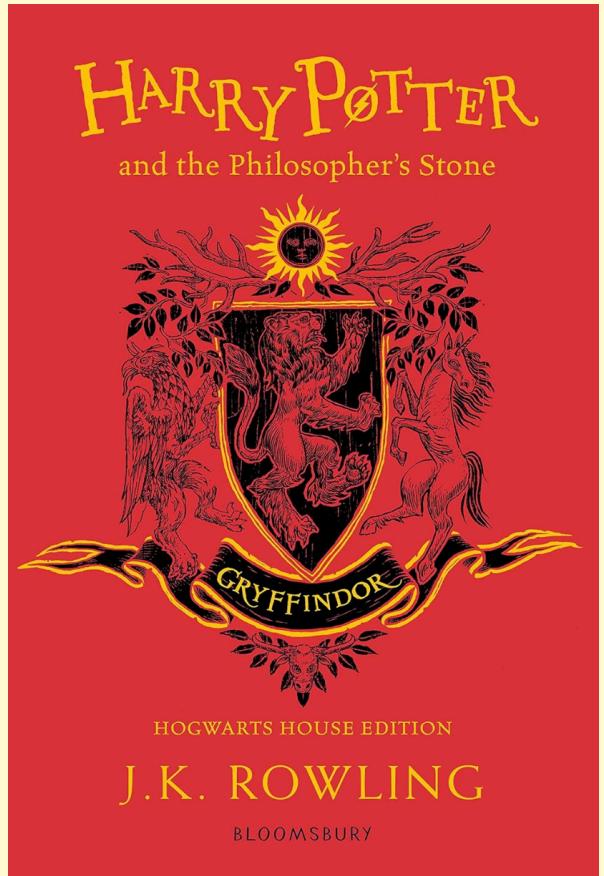
a convenient measure to evaluate the homogeneity level in a group is:

$$\text{Cohesion} = \sum_{x \in C} d_{cos}(x, c)$$

and finally, the total cohesion among groups can be calculated as:

$$\text{Total cohesion} = \sum_{r=1}^k \sum_{x \in C_r} d_{cos}(x, c_r)$$

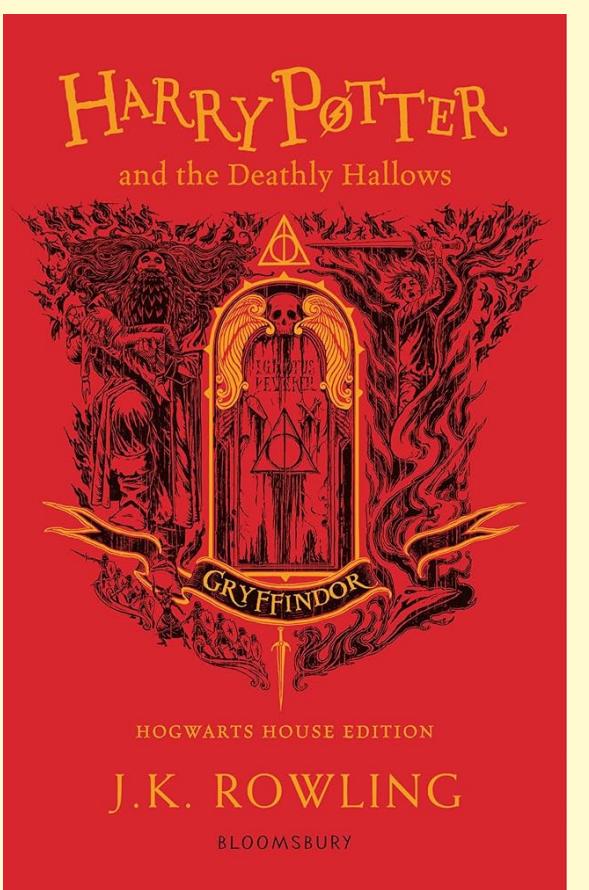
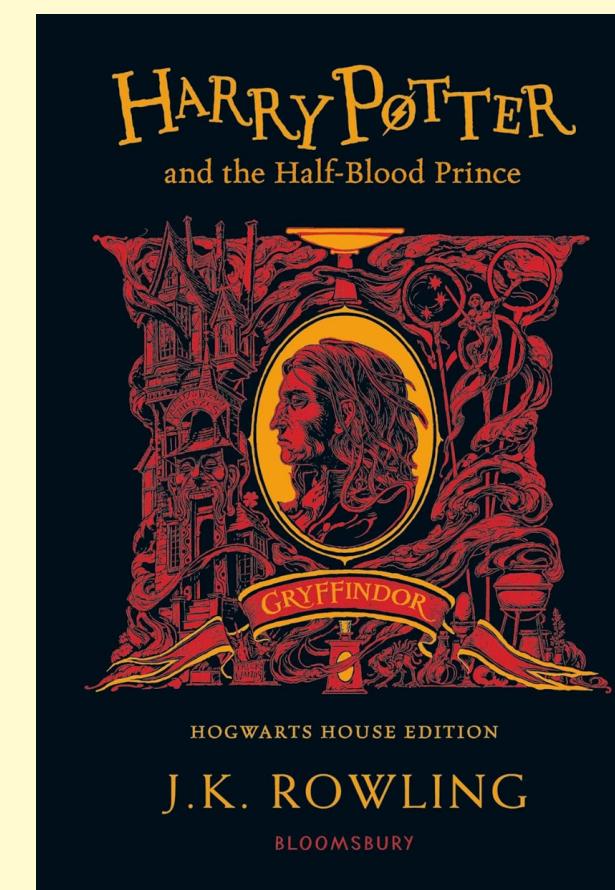
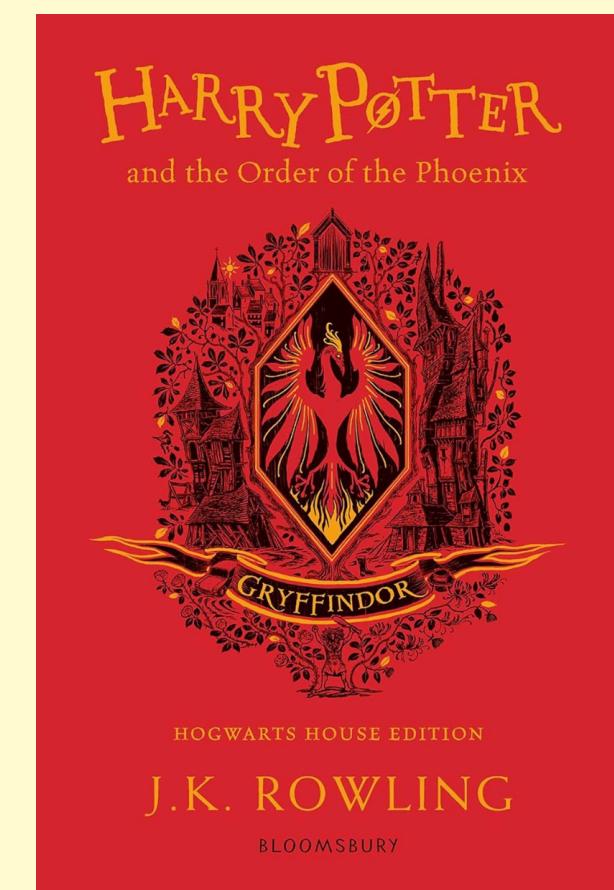
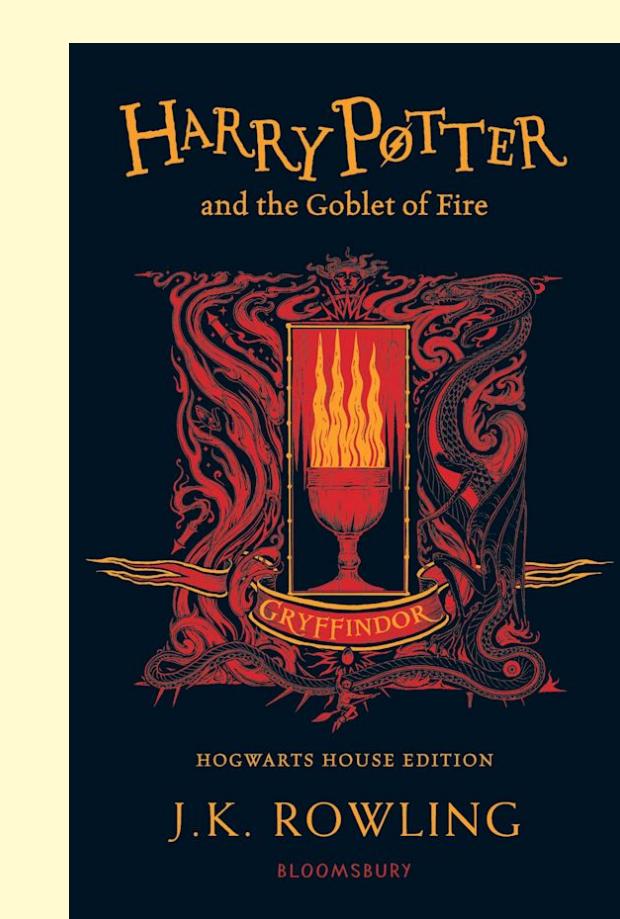
6. Clustering k-means



B.V →



A.V ←



7. Spectral Clustering

First we need to represent the data as a graph, so we calculate distances between the books. We define the matrices

$$W = (d_{cos}(x_i, x_j))_{ij} \quad \text{and} \quad D = diag(d_i)$$

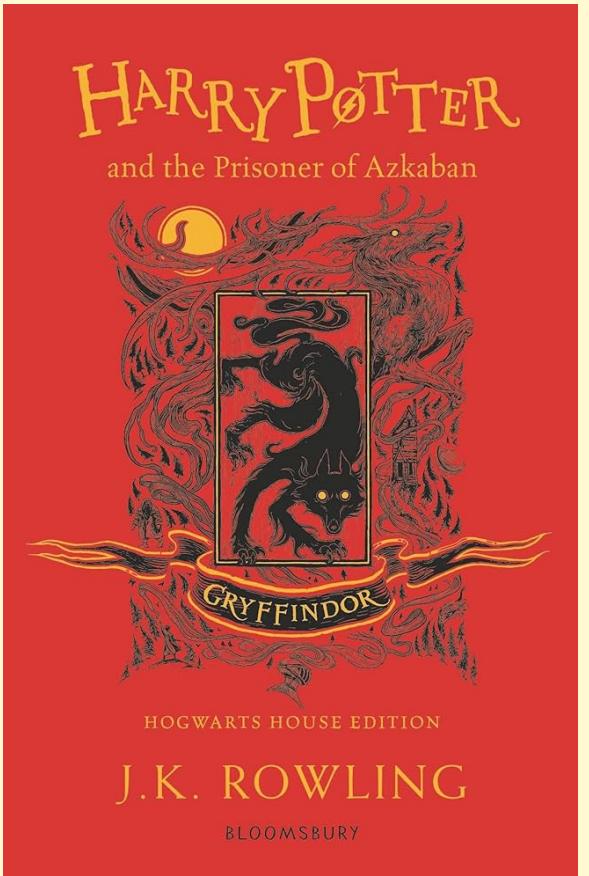
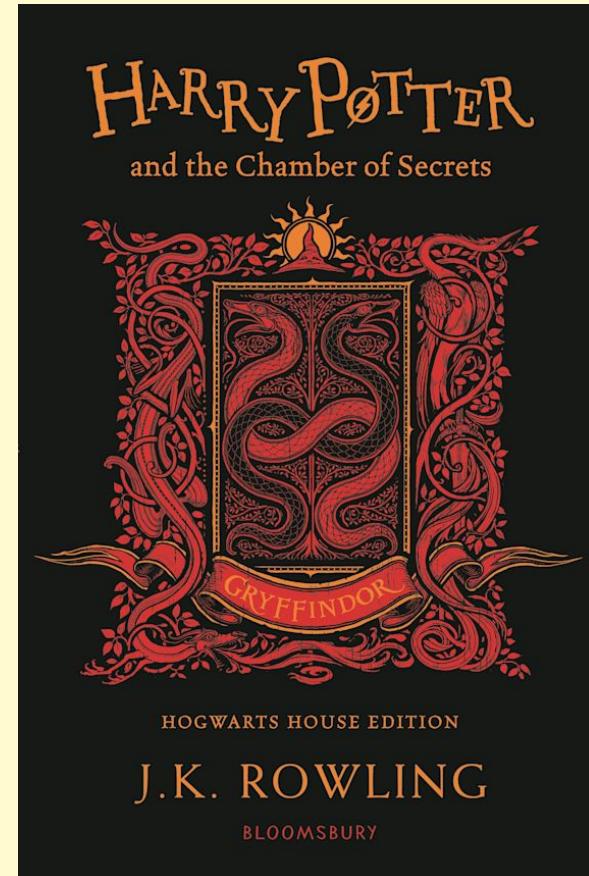
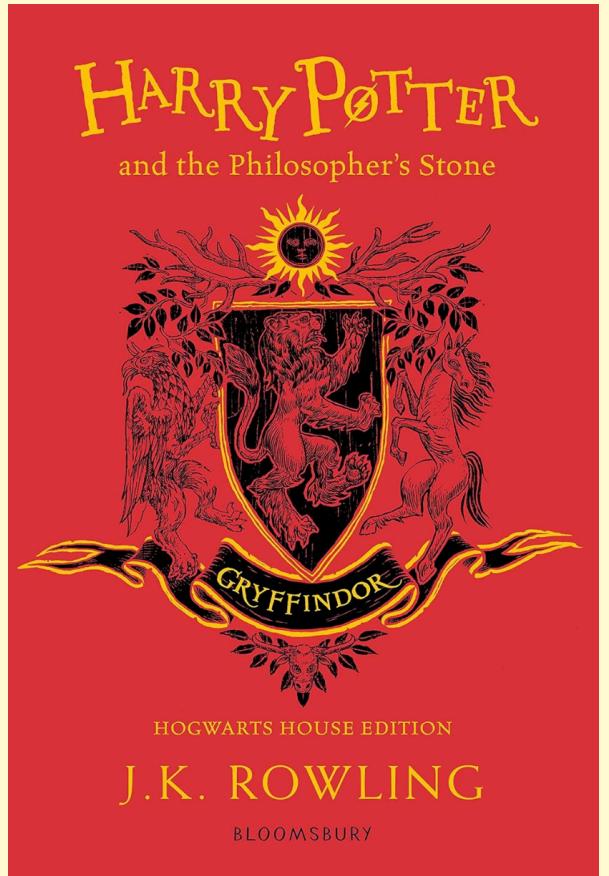
where D is a diagonal matrix, whose diagonal elements correspond to the degree associated to each book.

Then we compute:

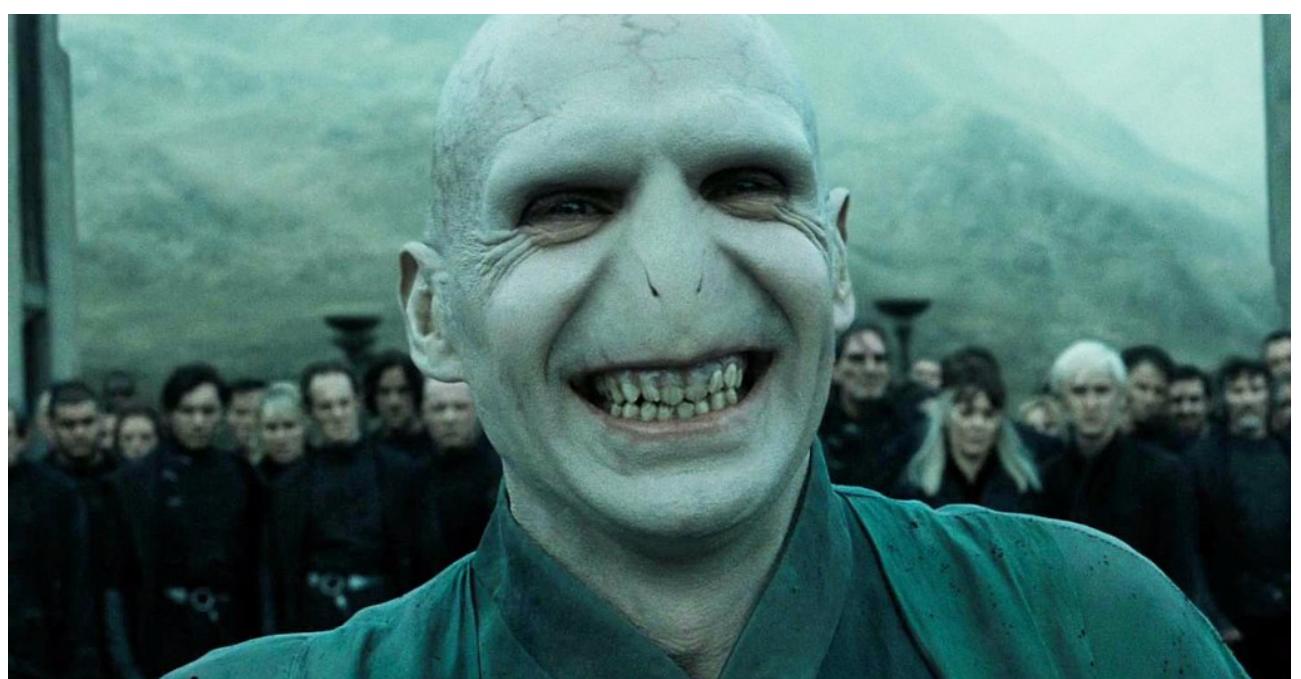
$$L = D - W \quad \text{and} \quad L_{sym} = D^{-1/2} L D^{-1/2}$$

We run a clustering algorithm (k-means) on the first k eigenvectors associated to the k smallest eigenvalues, where k is equal to the number of clusters desired. In this case we choose to divide the books in only two clusters.

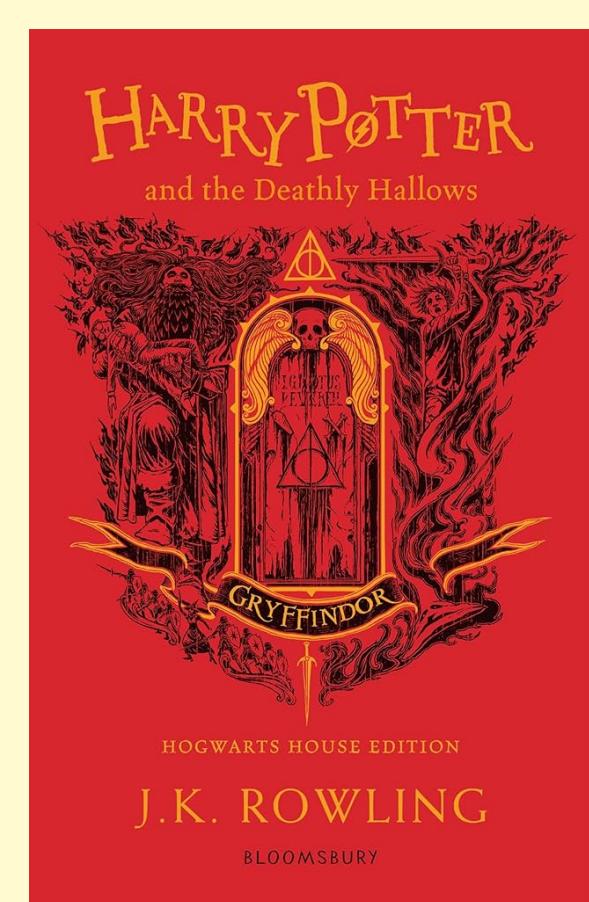
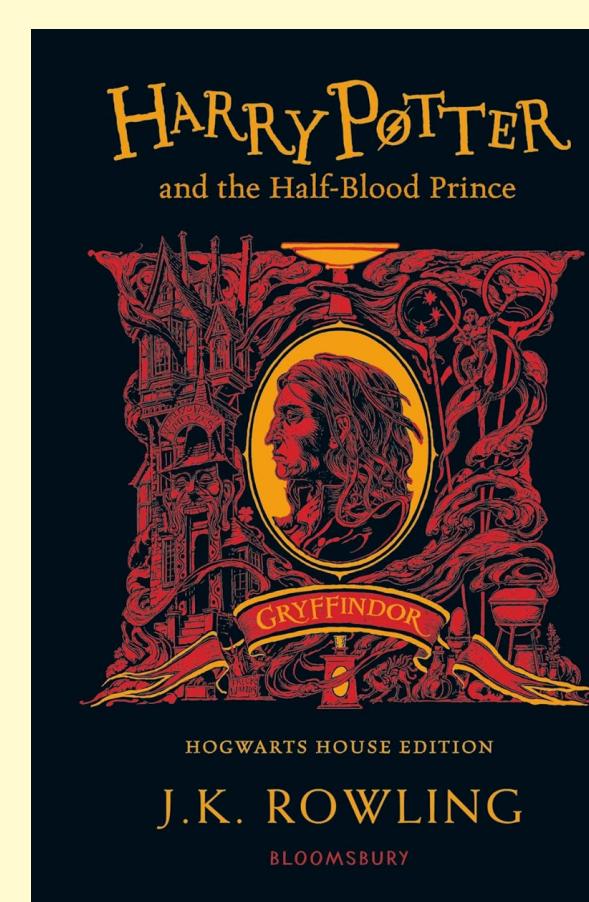
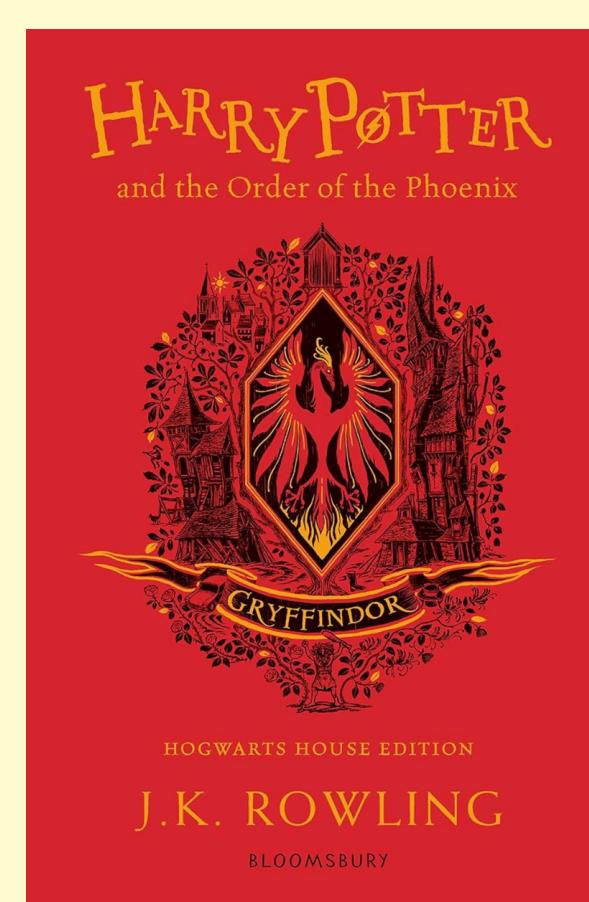
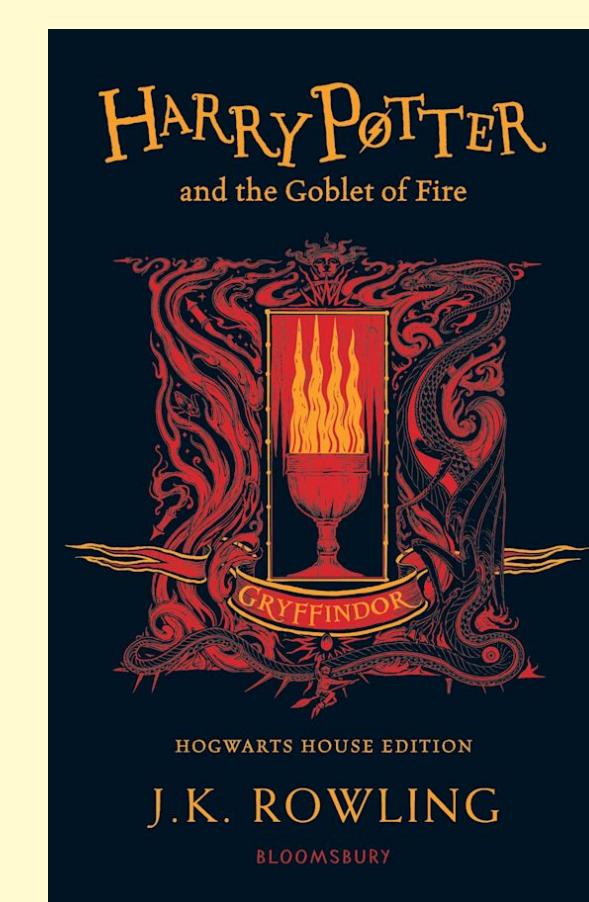
7. Spectral Clustering



B.V

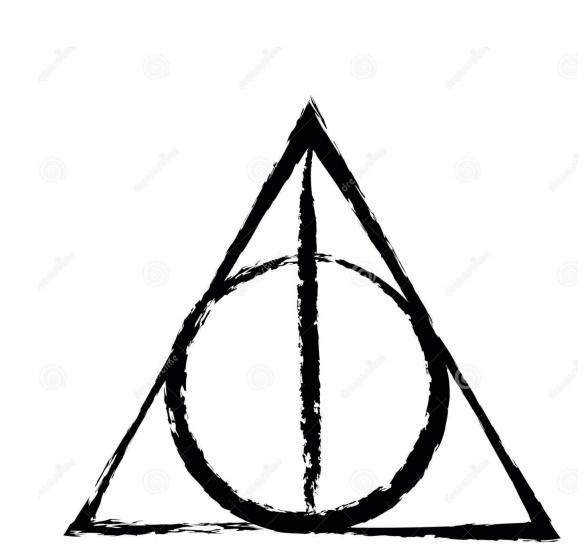


A.V



19 minutes later... (we hope)

THANK YOU FOR THE
ATTENTION



Enrico Carraro

Alex Cecchetto

Virginia Maffu