Emotion Classification of Classic Literature Using Natural Language Processing Models

Talia Duffy
Department of Statistics
University of Michigan
Ann Arbor, MI
taliagd@umich.edu

***Abstract*—We propose a novel implementation of the transformer-based text classification model, "Emotion English DistilRoBERTa-base" from HuggingFace. The texts of eight English novels are split into sentences, which are assigned one of seven emotion classes by the model. These methods support an in-depth emotional analysis of historical works of English literature.**

***Index Terms*—natural language processing, emotion classification, sentiment analysis, classic literature, English language**

I. Introduction

A. Motivation and Goals

Literature is an important mode of human expression. Literary works capture distinct qualities of the period in which they were written, like time capsules. But they simultaneously demonstrate the timelessness of human attitudes, perspectives, and emotions. Our understanding of these ideas and the study of literature generally can be supplemented or enhanced by natural language processing (NLP) techniques.

Each year, more classic novels are released from copyright and enter the public domain in the United States. For example, copyright expired for *The Great Gatsby* by F. Scott Fitzgerald in 2021 [1]. Project Gutenberg is an online, free collection of eBooks in the public domain – and a great source of text data from English literature [2].

This project uses an open-source HuggingFace text classification model to analyze the emotional content of eight classic English novels. This includes three sub-objectives: convert the full text of a novel into data for this purpose; successfully implement the model and obtain emotion classifications for the text data; and analyze the emotion classification results.

B. Model Background

The model chosen for this task is Jochen Hartmann's "Emotion English DistilRoBERTa-base" model from HuggingFace [3]. The data are the full texts of eight novels, downloaded as TXT files from Project Gutenberg. The model will be used to classify segments of each novel into one of seven emotion categories: anger, disgust, fear, joy, neutral, sadness, or surprise.

The Emotion English DistilRoBERTa-base model is trained on six datasets from varying sources, with each emotion equally represented. The model is a fine-tuned version of the "distilled" RoBERTa base model. This means that it is based on FacebookAI's RoBERTa model, which was released in 2019 as an improvement on Google AI's BERT language model [4]. For simpler use cases, a "distilled," or smaller and faster version of RoBERTa, was created and is also available on HuggingFace [5] [6]. Finally, the distilled RoBERTa was fine-tuned for the specific task of recognizing the seven listed emotions.

C. Prior Work

Most work on emotion classification and sentiment analysis of English text focuses on social media or internet data, such as posts from Twitter or movie reviews [7]. Transformer-based language models like RoBERTa generally have the best performance in this area, although small increases in accuracy can be demonstrated for specific tasks by combining RoBERTa with other deep learning techniques, like convolutional neural networks [8]. One paper performed emotion classification on books with data from Goodreads reviews [9].

Some recent work designed deep learning models specifically for emotion classification of classic literature, in English and in other languages [10] [11]. However, no work was found that leverages BERT or RoBERTa models for this purpose.

II. Methods

A. Data Processing

The first challenge was breaking up the novel into segments that allow for accurate and meaningful emotion classification. Each segment of the text (model input) is assigned one emotion label (model output). In a test run, we split the TXT file line by line; however, one line can contain an incomplete sentence or multiple sentences that convey contrasting emotions.

It is more natural to split the file sentence by sentence, which is a nontrivial task in itself. Parsing the file with the period character fails separate sentences that end in other punctuation marks. Further, the period character is used in abbreviations, acronyms, and ellipses. A sentence containing "Mrs." or "U.S.A." would be prematurely truncated. Accounting for all edge cases in all novels is intractable.

The Natural Language Toolkit (NLTK) Python library can help overcome these issues [12]. The library includes an unsupervised NLP model that can identify sentences in a string of text with less confusion from edge cases. For simpler

processing, we first split the TXT file into paragraphs, which are then split into sentences with the NLTK model. We save the paragraph number – indicated by integers increasing from 0 – to track each sentence's location in the novel.

B. Model Preparation and Implementation

For this project, we make predictions with the model's trainer, which allows greater control over training arguments and access to more detailed prediction results. The model and its tokenizer can be accessed with HuggingFace's Transformers library. We alter training arguments to allow a batch size of 16 for predictions. This helps decrease model runtime without exceeding the computational limits of a CPU.

A first pass of tokenized sentence data through the Emotion English DistilRoBERTa-base model revealed a major issue. The test novel projected prohibitively long runtimes in the range of 40-50 hours. For comparison, the test on line-by-line tokenized text from the same novel was completed in under four hours.

Transformer models like RoBERTa use self-attention to interpret meaning in text [13]. This requires the calculation of a $t \times t$ attention matrix for each tokenized input, where t is the number of tokens. Thus, the computational complexity of these models increases quadratically with input sequence length [14]. BERT and RoBERTa will not process inputs longer than 512 tokens, and even this length is too much for a 2020 MacBook to handle. Classic novels often contain long sentences that exceed reasonable token lengths. Indeed, the longest sentence in the test novel is 629 tokens, and many sentences are longer than 100 tokens.

To address this, we divide sentences into two lists: "short sentences" that can feasibly be sent through the model and "long sentences" that may cause runtime issues. As mentioned previously, the line-by-line test finished in an acceptable four hours, and the longest line is 33 tokens. Therefore, less than or equal to 40 tokens is a reasonable length requirement for short sentences. The differences in the classification process for long and short sentences are as follows.

a) Short Sentences: We tokenize short sentences with dynamic padding, which means each tokenized sentence is padded to the longest token length in the batch. We set the trainer to make predictions in batches of 16 sentences. The model input is the list of tokenized short sentences.

Suppose there are n_s short sentences. The model output is an $n_s \times 7$ array where each row corresponds to a sentence and each column is the logit value for one of the seven emotion labels, numbered 0 through 6. The label with the highest logit is chosen for each sentence, and each label is matched with its named emotion class. For easier interpretation, the vector of logits y is transformed into a vector of probabilities with the softmax function:

$$\sigma(y_i)_i = \frac{e^{y_i}}{\sum_{j=0}^{6} e^{y_j}}$$

which ensures that the entries in the probability vector are positive and sum to 1.

b) Long Sentences: Any sentence longer than 40 tokens is "chunked," or split into subsequences of 40 tokens or less. This prevents the attention matrices from growing prohibitively large.

To make predictions, we iterate through the list of long sentences, each now represented by its own list of sentence "chunks." One iteration processes one long sentence. The list of chunks is tokenized without padding. This becomes the model input, and the model output is an array, similar to before, where each row corresponds to one chunk and each column corresponds to the logit value for one of the labels.

We use a separate trainer that does not implement batching to make predictions for long sentences. This is because there are typically less than 16 chunks in one long sentence, so it is meaningless to process the chunks in batches of 16.

As before, the logits for each chunk are transformed into probabilities with the softmax function. Then, the average probability for each label is calculated over the chunks. Consider a long sentence that is split into 3 chunks. Each of the k = 1, 2, 3 chunks has a probability associated with label 0, p_{0_k}. Therefore the final probability for label 0 in the long sentence is:

$$\frac{1}{3}\sum_{k=1}^{3} p_{0_k}$$

The label with the largest mean probability is assigned to the sentence, which is mapped to the appropriate emotion class.

c) Recombination: The results for long and short sentences are combined in one dataset where each row corresponds to one sentence. The rows are organized so the sentences are in the same order as in the novel. Saved in the dataframe are:

- Full sentence text
- Paragraph number
- Predicted label 0-6
- Corresponding emotion class, one of {anger, disgust, fear, joy, neutral, sadness, surprise}
- Probability of the predicted label
- One column each for the probability of {anger, disgust, fear, joy, neutral, sadness, surprise}

III. Results

Once the results are obtained and saved, we proceed with emotional analysis. These methods were primarily tested with Oscar Wilde's *The Picture of Dorian Gray*, chosen for its relatively short length and well-known demonstration of negative emotions. We also analyzed seven other classic English novels; some results for these novels are in the appendix.

We produce bar charts to display the frequency of each emotion in a novel. As seen in "Fig. 1", the most common non-neutral labels in *Dorian Gray* were disgust and fear. Qualitatively, this is accurate; the titular character fears growing older and losing his beauty, which causes him to descend into vice and madness.

For every novel, neutral was the most frequently assigned label. This is expected, as many sentences in literature are

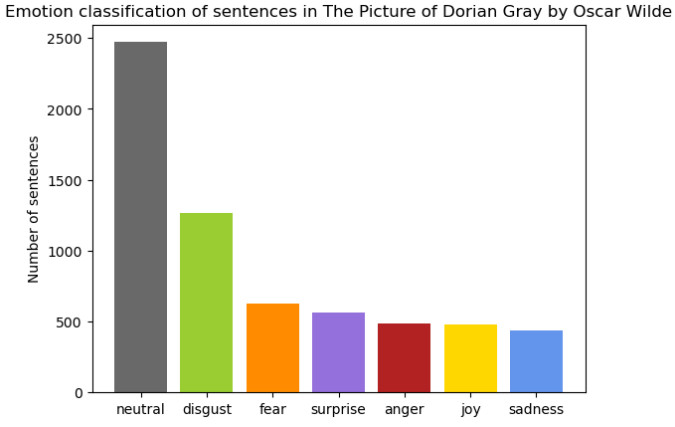

Fig. 1. Summary of results for the test novel.

more descriptive of actions or setting than they are emotional. *Frankenstein* by Mary Shelley had the lowest percentage of neutrally classified sentences at 20.9%. *The Adventures of Huckleberry Finn* by Mark Twain was the most neutral at 55.3%. We also take interest in which sentences were classified with the highest probability; in *Dorian Gray*, this is sentence 3,346 in paragraph 771, classified as fear with probability 99.4%:

"A cry of terror broke from Dorian Gray's lips, and he rushed between the painter and the screen."

For a more nuanced look, we produce line graphs that track the probabilities of each emotion as the novel progresses. Probabilities can be considered a measure of the strength of each emotion in a sentence. We take averages in each paragraph for visual simplicity, and square probabilities to accentuate spikes in emotion.

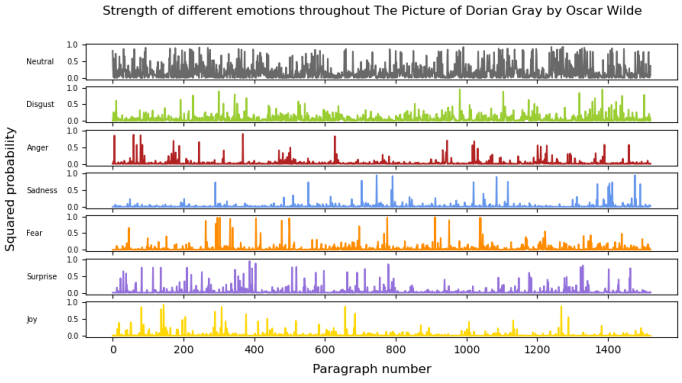

Fig. 2. Tracking emotions through the test novel.

This plot draws attention to significant moments in a novel. Note in "Fig. 2" the spike of disgust around paragraphs 970-1000. In this section of the novel, Dorian Gray murders another character. Several sentences depict gruesome details of the death. In a similar plot generated for *Alice's Adventures in Wonderland* by Lewis Carroll, a spike in anger corresponds to the first appearance of the Queen of Hearts, who vehemently demands: "Off with his head!"

One major shortcoming of this analysis is the lack of a rigorous estimate for accuracy. It is time consuming for a human coder to read and classify each sentence of multiple novels. Further, literature is subjective, so two human coders might have different interpretations of the same sentence.

The model's capability for this use case is weakened because each sentence is processed in isolation. In a novel, the emotions conveyed by a sentence are often dependent on sentences around it. This problem is exacerbated by the chunking strategy for long sentences, since each chunk is removed from the context of the entire sentence. An improved strategy might implement more advanced attention mechanisms within paragraphs, but would likely be quite slow.

IV. Conclusion

The combination of publicly available text data and NLP technology allows historical literary works to be analyzed in new ways. We obtained text data from eight classic English novels and assigned each sentence one of seven emotion labels with the Emotion English DistilRoBERTa-base model from HuggingFace. The results can lead to interesting insights about a novel's emotional content.

One area for improvement is increasing the model's ability to understand emotions in the context of surrounding sentences or paragraphs without inflating runtime. Another is the development a rigorous assessment the model's accuracy on literary text data.

However, the analysis of literature has never been an exact science. These results are supplemental, and should be viewed from a critical, thoughtful, and human perspective.

References

[1] N. Ulaby, "Party Like It's 1925 On Public Domain Day (Gatsby And Dalloway Are In)," NPR, 2021. https://www.npr.org/2021/01/01/951171599/party-like-its-1925-on-public-domain-day-gatsby-and-dalloway-are-in (accessed Dec. 05, 2025).

[2] M. Hart, "The Project Gutenberg Mission Statement," Project Gutenberg, Jun. 20, 2004. https://www.gutenberg.org/about/background/mission_statement.html (accessed Dec. 05, 2025).

[3] J. Hartmann, "Emotion English DistilRoBERTa-base," HuggingFace, 2022. https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," ArXiv, Oct. 11, 2018. https://arxiv.org/abs/1810.04805

[5] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv.org, Jul. 26, 2019. https://arxiv.org/abs/1907.11692

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv.org, 2019. https://arxiv.org/abs/1910.01108

[7] X. Liu et al., "Emotion classification for short texts: an improved multi-label method," vol. 10, no. 1, Jun. 2023, doi: https://doi.org/10.1057/s41599-023-01816-6.

[8] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Improving sentiment classification using a RoBERTa-based hybrid model," Frontiers in Human Neuroscience, vol. 17, p. 1292010, Dec. 2023, doi: https://doi.org/10.3389/fnhum.2023.1292010.

[9] E. R. Luţan and C. Bădică, "Emotion-Based Literature Book Classification Using Online Reviews," Electronics, vol. 11, no. 20, p. 3412, Oct. 2022, doi: https://doi.org/10.3390/electronics11203412.

[10] J. Yu and C. Qi, "Machine Learning-Based Sentiment Analysis in English Literature: Using Deep Learning Models to Analyze Emotional and Thematic Content in Texts," IEEE Access, vol. 13, pp. 65997–66008, Jan. 2025, doi: https://doi.org/10.1109/access.2025.3553386.
[11] Q. Gao and J. Huang, "Design and implementation of classical literature sentiment analysis system based on ensemble learning and graph neural network," International Journal of Cognitive Computing in Engineering, vol. 6, pp. 603–616, May 2025, doi: https://doi.org/10.1016/j.ijcce.2025.05.004.
[12] "NLTK : nltk.tokenize.punkt module," www.nltk.org. https://www.nltk.org/api/nltk.tokenize.punkt.html
[13] A. Vaswani et al., "Attention Is All You Need," Cornell University, Jun. 12, 2017. https://arxiv.org/abs/1706.03762
[14] S. Mehta, H. Rangwala, and N. Ramakrishnan, "Compact Multi-Head Self-Attention for Learning Supervised Text Representations," arXiv.org, Aug. 10, 2020. https://arxiv.org/pdf/1912.00835

APPENDIX

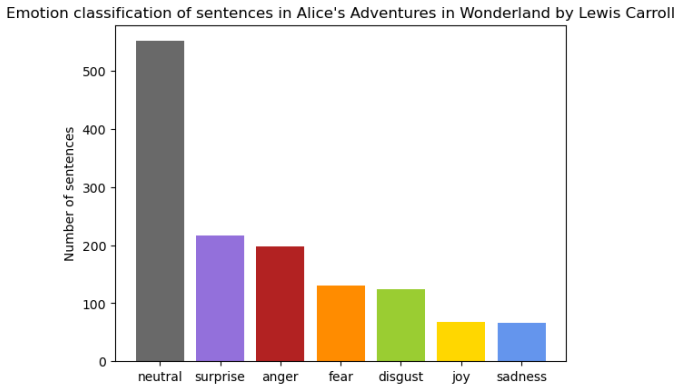

Fig. 3. Results for *Alice's Adventures in Wonderland.*

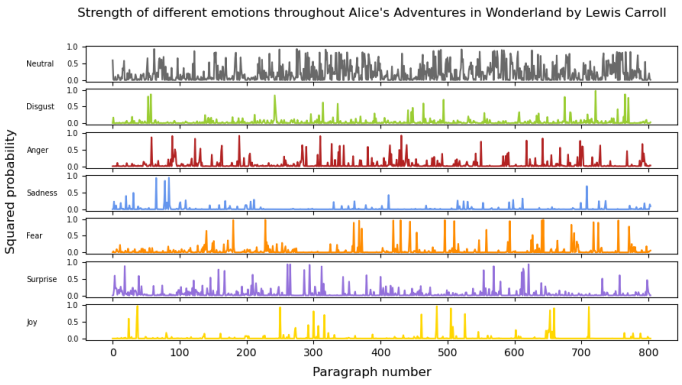

Fig. 4. Results for *Alice's Adventures in Wonderland.*

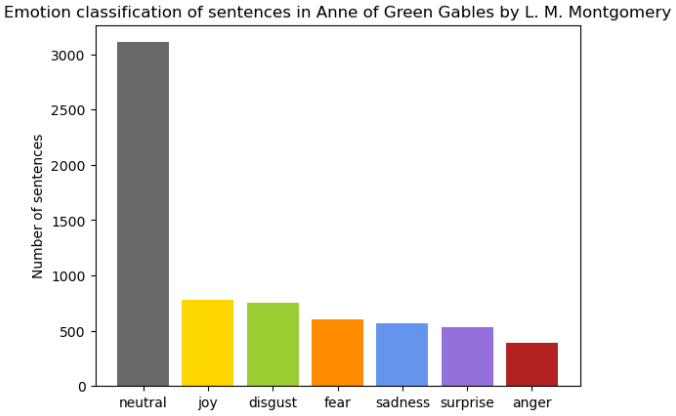

Fig. 5. Results for *Anne of Green Gables.*

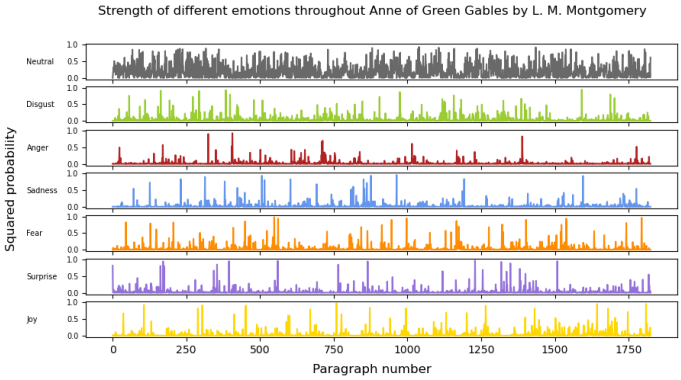

Fig. 6. Results for *Anne of Green Gables.*

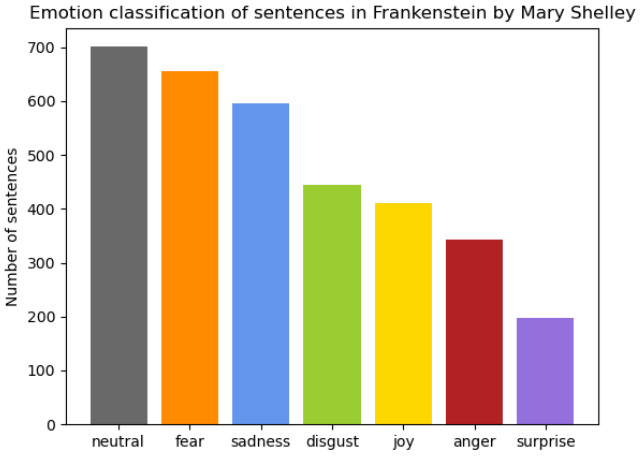

Fig. 7. Results for *Frankenstein; or, The Modern Prometheus.*

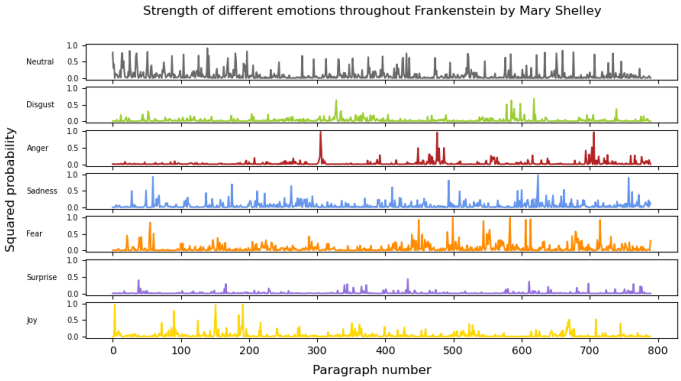

Fig. 8. Results for *Frankenstein; or, The Modern Prometheus.*

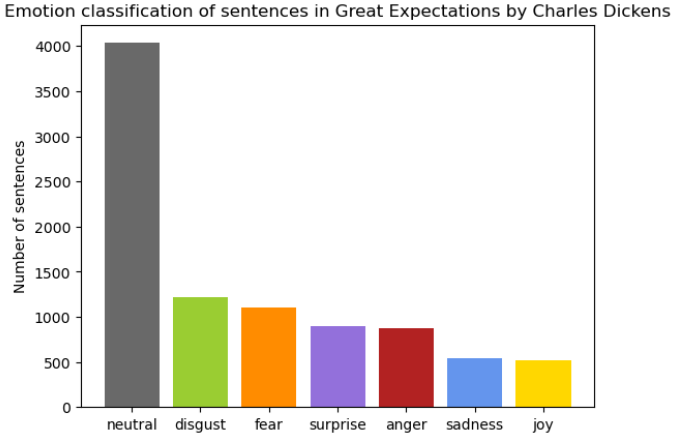

Fig. 9. Results for *Great Expectations.*

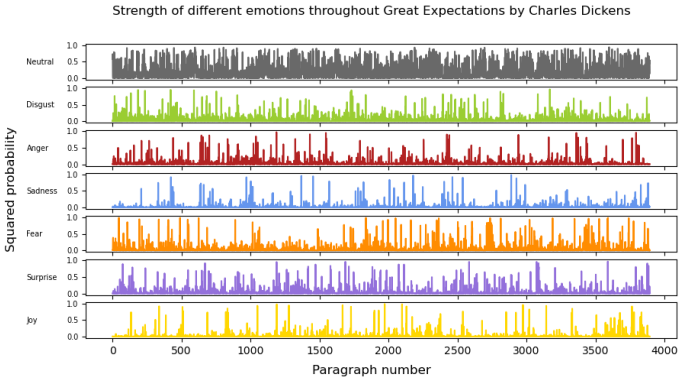

Fig. 10. Results for *Great Expectations.*

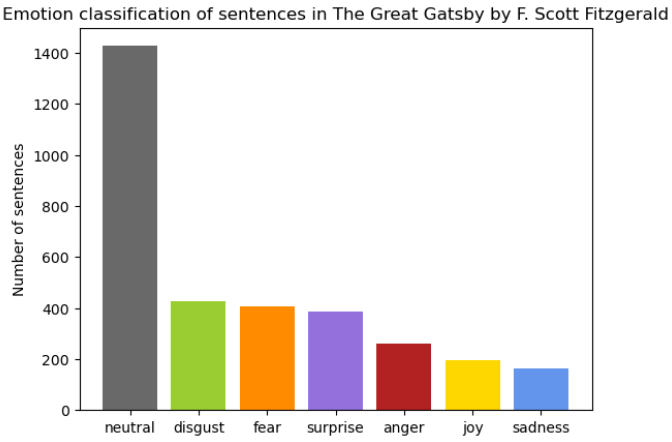

Fig. 11. Results for *The Great Gatsby.*

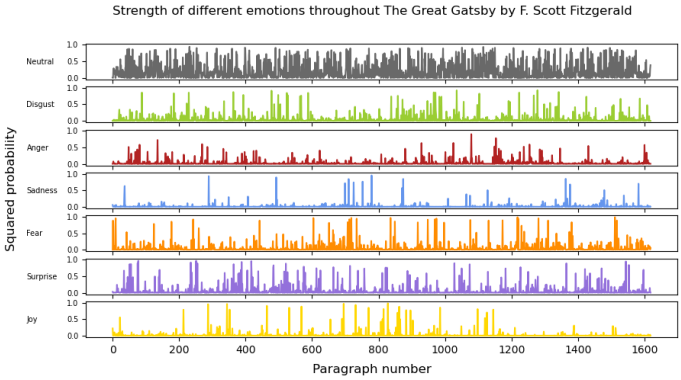

Fig. 12. Results for *The Great Gatsby.*

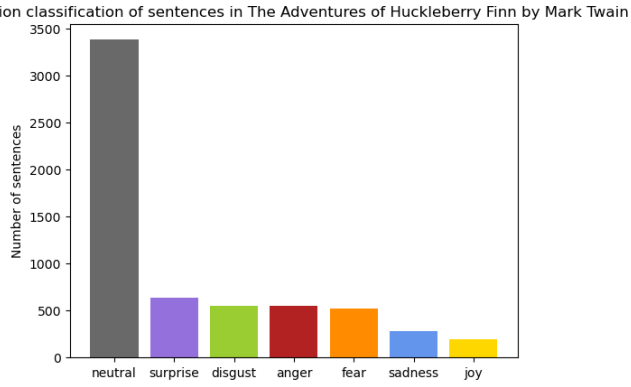

Fig. 13. Results for *The Adventures of Huckleberry Finn.*

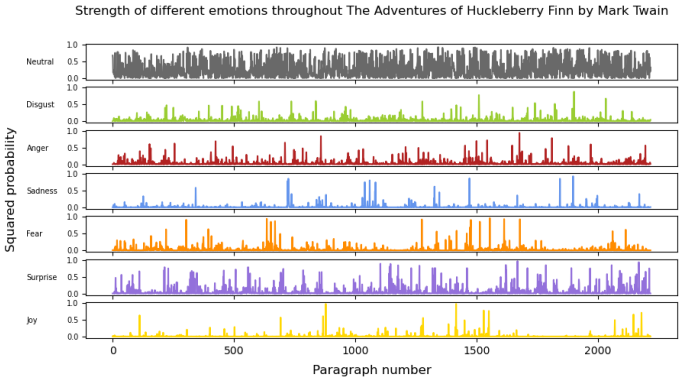

Fig. 14. Results for *The Adventures of Huckleberry Finn.*

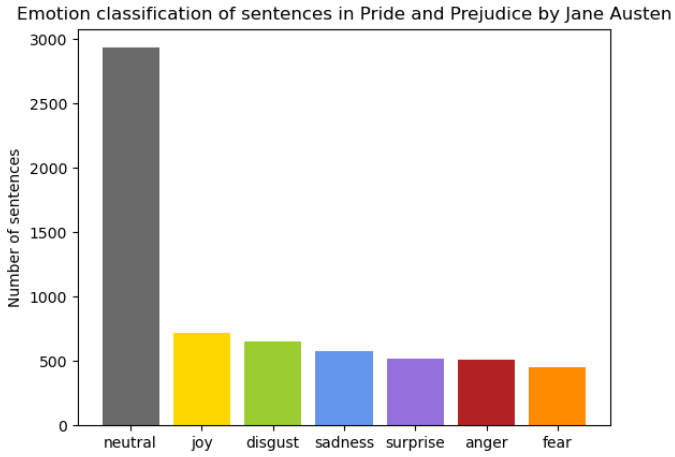

Fig. 15. Results for *Pride and Prejudice.*

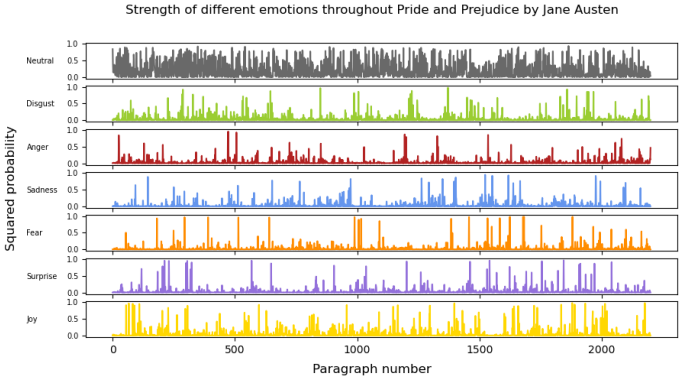

Fig. 16. Results for *Pride and Prejudice.*