

# Glossier Sentiment Analysis

Analyzing reviews from  
Glossier's recent  
product relaunch



# The New Balm Dotcom

Glossier relaunched their customer favorite product on February 1st, 2023, introducing a new formula and new packaging design

# Goals

- Find overall sentiment towards newly launched product
- See what exactly customers are saying

# Methodology

- Data collection and analysis done using: Python, R, Openrefine, ChatGPT API
- Python
  - Gathering initial data
  - Converting JSON to CSV
  - Analysis and visualization creation
- Openrefine
  - Clean, split up, and merge datasets
- R
  - Create wordclouds and get word counts
- ChatGPT API
  - Perform sentiment analysis on reviews (using Python)

# Gathering Data

```
sephora_review_dict = []
```

```
for offset in range(10):
```

```
    url = f'https://api.bazaarvoice.com/data/reviews.json?
    Filter=contentlocale%3Aen*&Filter=ProductId%3AP504525&So
    rt=SubmissionTime%3Adesc&Limit=100&Offset
```

```
offset*100}&Include=Products%2CComments&Stats=Reviews&passk
ey=calXm2DyQVjcCy9agq85vmTJv5ELuuBCF2sdg4BnJzJus&apiversion=5.4&Locale=en_
US'
```

```
    headers = {'Content-Type' : 'application/json'}
```

```
    res = requests.get(url, headers = headers)
```

```
    res_json = res.json()
```

```
    for review in res_json['Results']:
```

```
        review_data = {
```

```
            'ProductId' : review['ProductId'],
```

```
            'Title' : review['Title'],
```

```
            'ReviewText' : review['ReviewText'],
```

```
            'SubmissionTime' : review['SubmissionTime'],
```

```
            'Rating' : review['Rating']
```

```
        }
```

```
    sephora_review_dict.append(review_data)
```

```
with open('sephora_bdc_reviews.json','a') as f:
```

```
    json.dump(sephora_review_dict, f)
```

```
with open('sephora_bdc_reviews.json', encoding = 'utf-8') as sephora_api:
```

```
    df = pd.read_json(sephora_api)
```

```
df.to_csv('sephora_bdc_reviews.csv', encoding = 'utf-8')
```

```
glossier_review_dict = []
```

```
for page in range(51):
```

```
    url = f'https://api-
```

```
cdn.yotpo.com/v1/widget/xAJLIqkdZjK3FOIgCqiXbx1OtYADZxE80pWNsAW/products/
7985671504117/reviews.json?sort=votes_up&per_page=150&page={page+1}'
```

```
    headers = {'Content-Type' : 'application/json'}
```

```
    res = requests.get(url, headers = headers)
```

```
    res_json = res.json()
```

```
    for review in res_json['response']['reviews']:
```

```
        review_data = {
```

```
            'Rating' : review['score'],
```

```
            'Title' : review['title'],
```

```
            'ReviewText' : review['content'],
```

```
            'SubmissionDate' : review['created_at'],
```

```
            'Sentiment' : review['sentiment']
```

```
        }
```

```
    glossier_review_dict.append(review_data)
```

```
with open('glossier_bdc_reviews.json','a') as f:
```

```
    json.dump(glossier_review_dict, f)
```

```
with open('glossier_bdc_reviews.json', encoding = 'utf-8') as glossier_api:
```

```
    df = pd.read_json(glossier_api)
```

```
df.to_csv('glossier_reviews.csv', encoding = 'utf-8')
```

# Performing Sentiment Analysis

1

```
def analyze_sentiment(review):
    res = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        max_tokens=5,
        messages=[
            {"role": "system", "content": system_setting},
            {"role": "user", "content": f"""
                Give a sentiment analysis of the following review for a chapstick. Only analyze the sentiment towards the product.
                A review has a title and body. Output should not include any alphabetical characters.
                Do not respond with any text, only a singly float value between -1.0 and 1.0: {review}
            """}
        ]
    )
    res_text = res.choices[0].message.content
    print(res_text)
    scores = re.findall(r"[-+]?(?:\d*\.\d+)", res_text)
    if len(scores) > 0:
        return float(scores[0])
    else:
        return None
```

2

```
sentiments = []

with open('bdc_2023.csv', 'r') as file:

    reader = csv.DictReader(file)

    for row in reader:

        sentiment = analyze_sentiment(f"{row['Title']}, {row['ReviewText']}")
        sentiments.append(sentiment)
```

3

```
print(len(sentiments))
bdc_23 = pd.read_csv('bdc_2023.csv')

bdc_23['Sentiment'] = sentiments

bdc_23.head(10)

bdc_23.to_csv('bdc_2023_sentiment.csv')
```

# Example Code in R

```
bdc_sentiment_neg_words <- tibble(line=1:nrow(bdc_sentiment_neg), text = bdc_sentiment_neg$ReviewText) %>% unnest_tokens(word,
text, token = 'ngrams', n = 4 ) %>%
  anti_join(custom_stop_words) %>%
  count(word, sort = TRUE)

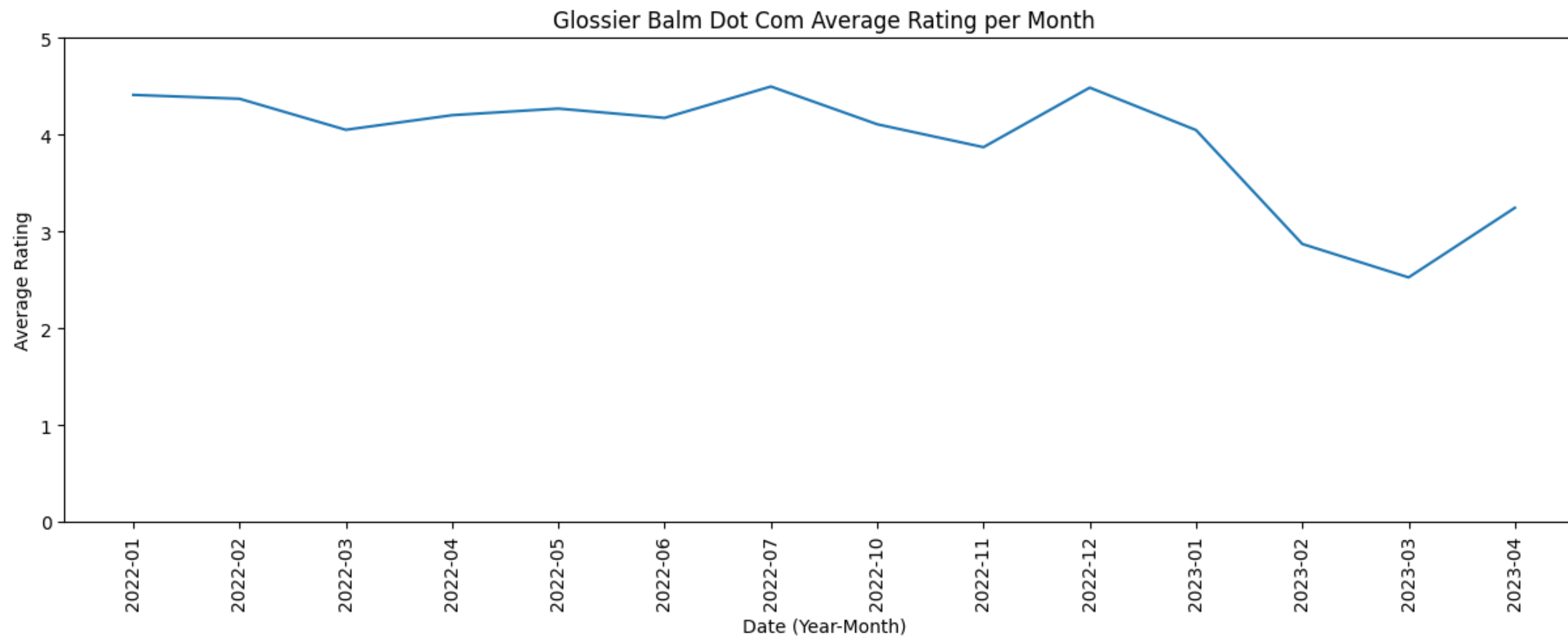
print(bdc_sentiment_neg_words)

set.seed(1234)
wordcloud(words = bdc_sentiment_neg_words$word, freq = bdc_sentiment_neg_words$n, min.freq = 1,
  max.words=25, random.order=FALSE, rot.per=.35,colors=brewer.pal(8,"Dark2"))
```

# Wordcloud in R vs. Python

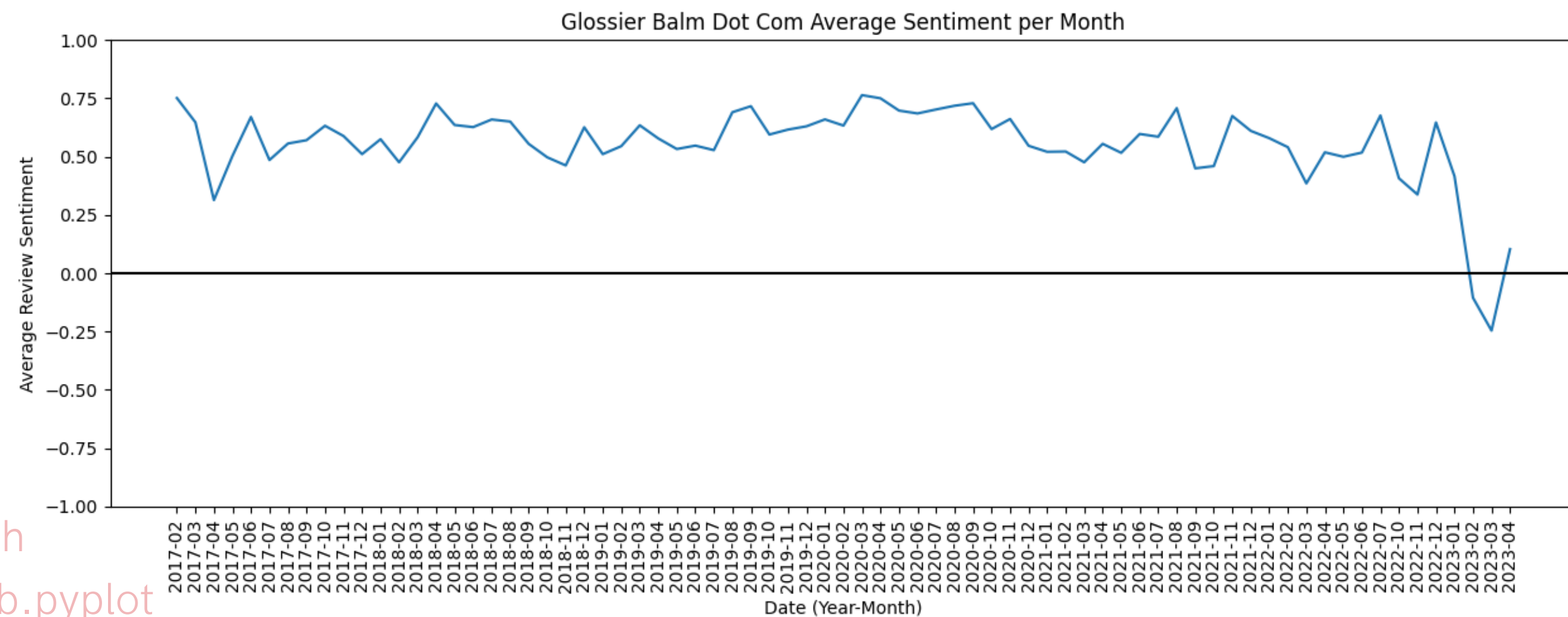


**Analysis**



# Average Rating and Sentiment

| SubmissionDate | Rating   | Sentiment |
|----------------|----------|-----------|
| 2017           | 4.252927 | 0.570036  |
| 2018           | 4.313208 | 0.593990  |
| 2019           | 4.356427 | 0.602724  |
| 2020           | 4.556263 | 0.668546  |
| 2021           | 4.308188 | 0.555292  |
| 2022           | 4.320246 | 0.550438  |
| 2023           | 2.944713 | -0.068251 |



Made with  
matplotlib.pyplot

# Negative Reviews

2023

but the new formula  
this new formula is  
with the old formula  
was so excited to  
it makes my lips  
new formula is so  
not moisturizing at all  
sits on top of  
as good as the  
i loved the old  
as the old one  
i love the new  
i miss the old  
i really wanted to  
i will not be  
dried out my lips  
the new applicator is  
loved the old formula  
leaves my lips feeling  
miss the old formula  
please bring back the  
the old formula was  
back the old formula  
bring back the old  
new formula is not  
of the old formula  
i was so excited  
as the old formula  
the new applicator but  
bring back the original  
dries out my lips  
the old formula back  
new formula is awful  
like the old formula  
the old formula the  
they changed the formula  
good as the old  
this used to be  
i have to reapply  
of the new formula  
with the new formula  
the new formula and  
i have been a

Average Sentiment: **-.634**

Average Rating: **1.763**

Total Reviews: **1462**

## FREQUENCIES

'disappointed': **209**

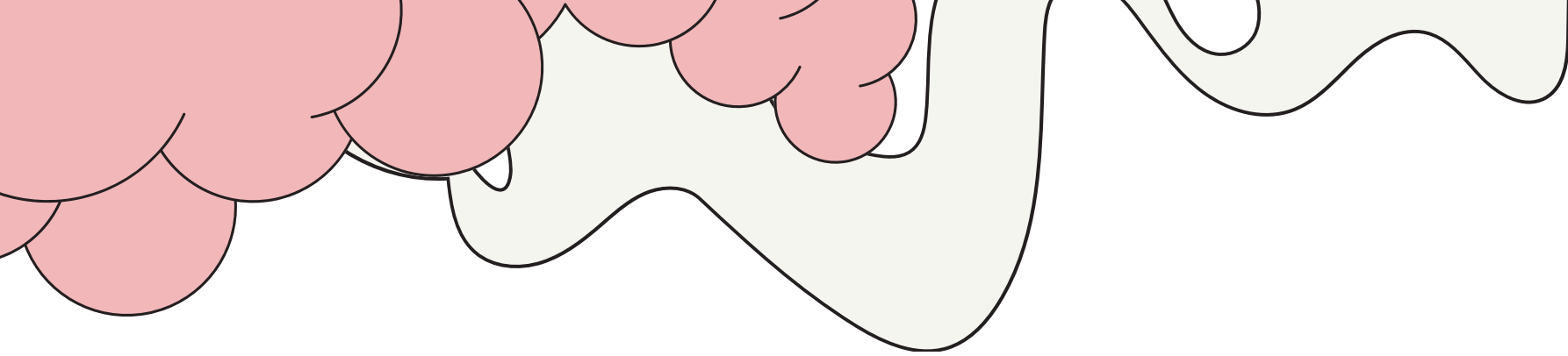
'dry' or a variation of the  
word: **598**

'cheap': **36**

'terrible': **83**

'awful': **67**

'horrible': **62**



# Positive Reviews

2023

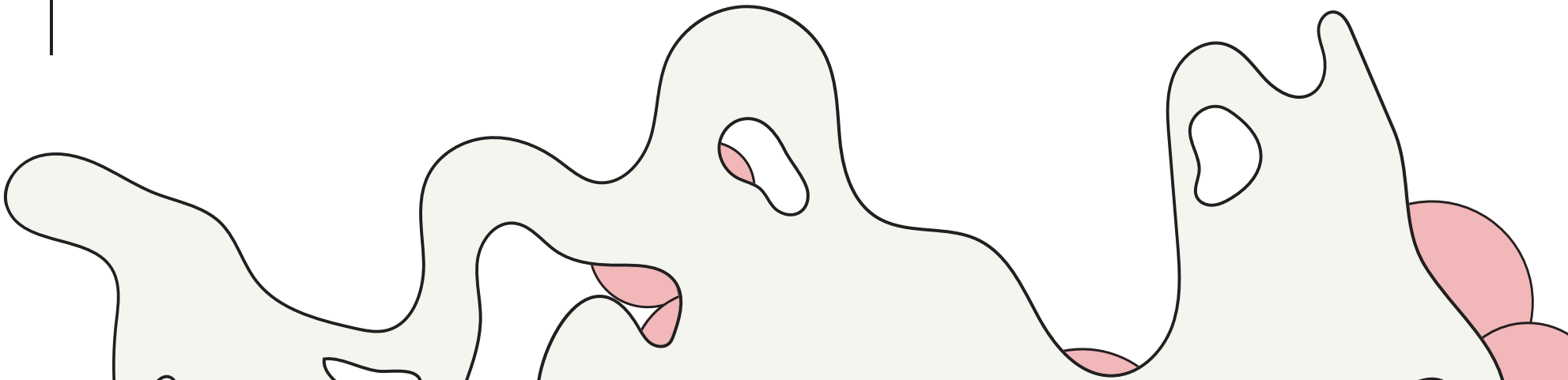
Average Sentiment: .7648

Average Rating: 4.672

Total Reviews: 992

## FREQUENCIES

- 'love': 539
- 'applicator': 178
- 'moisturizing': 110
- 'hydrating': 104
- 'packaging': 133
- 'perfect': 73



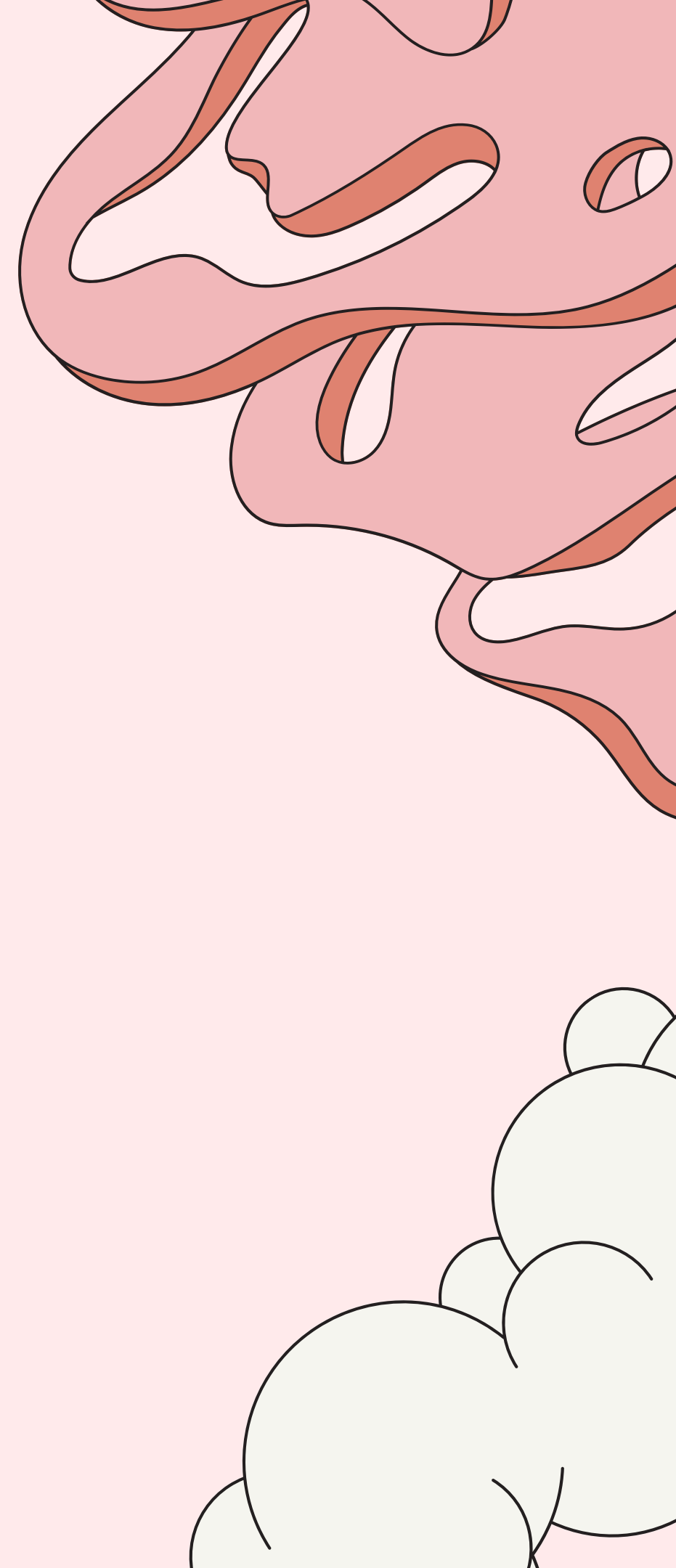
the new formula and packaging  
the new formula and i  
i have been using balm love the new packaging and  
love the new applicator i  
it smells so good and  
love the new formula and  
never tried the old formula i love this lip balm  
love the new applicator and  
i love the new packaging  
little goes a long way  
i love the new applicator  
i am in love with  
better than the old one  
i like the new formula  
a little goes a long  
to try the new formula  
i love the new formula  
i absolutely love the new  
i love it so much  
it makes my lips feel  
and i love the new  
and it keeps my lips

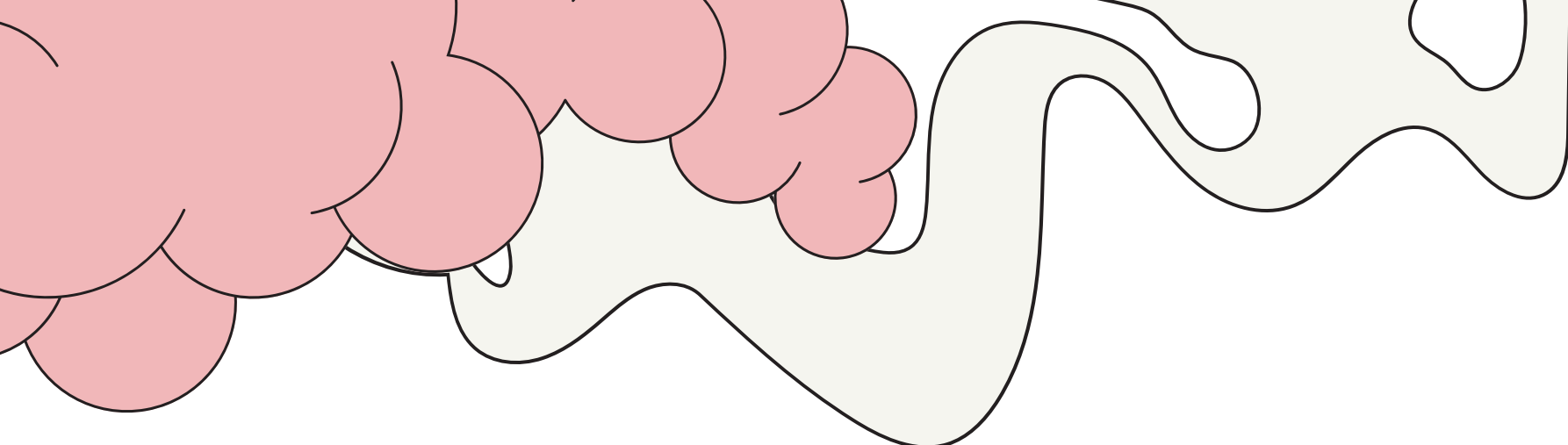
# ADDITIONAL STATS

Average Sentiment for reviews with  
'applicator':  $-.1634$

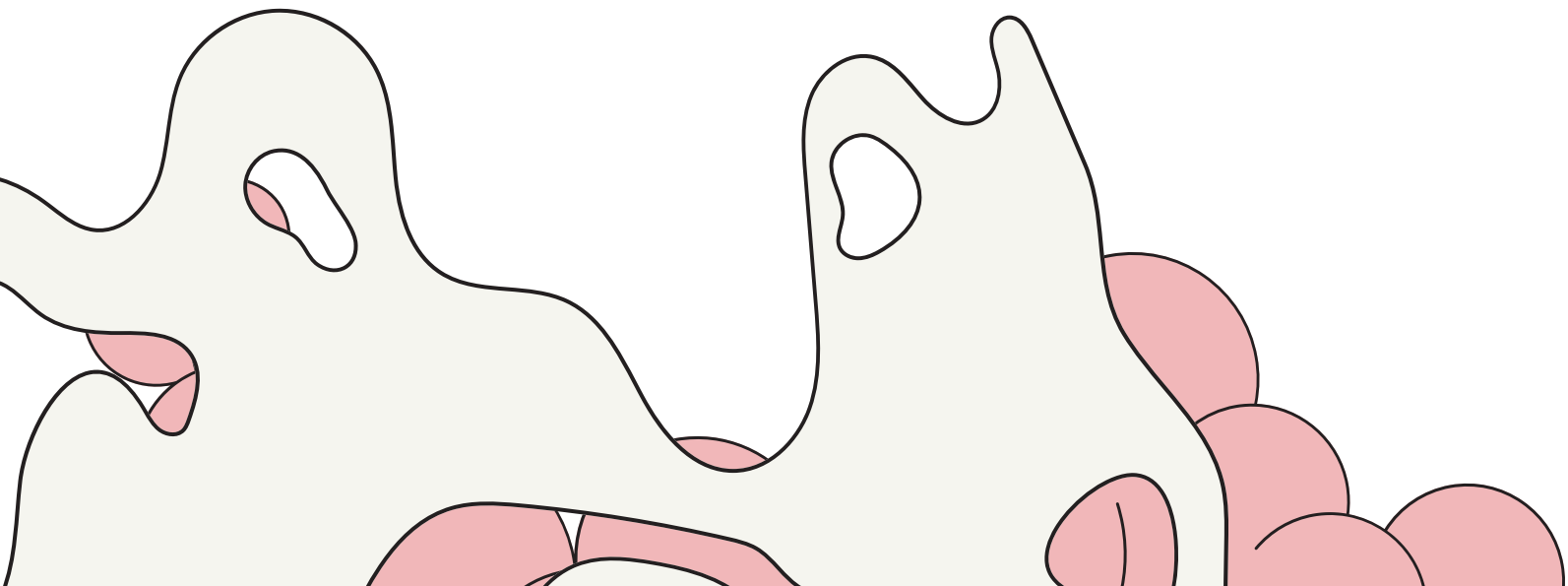
Average Sentiment for reviews with 'new  
formula':  $-.4074$

Average Sentiment for reviews with 'old  
formula':  $-.4388$





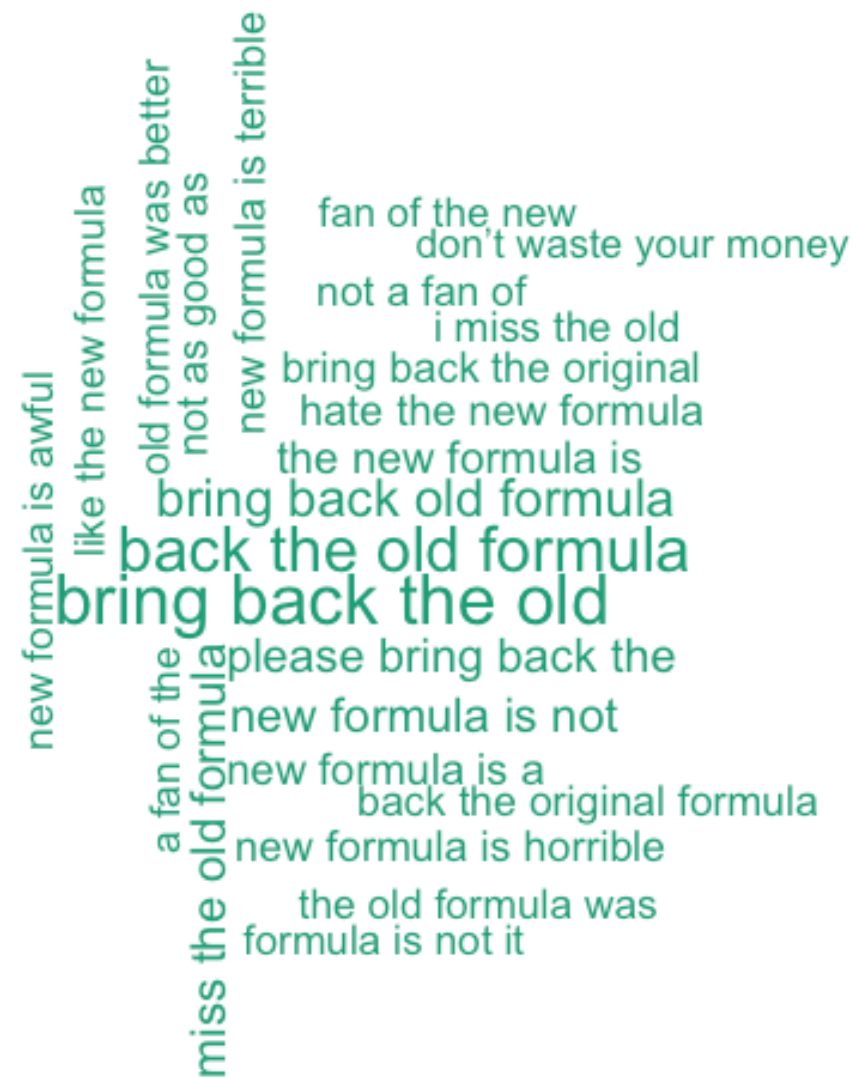
# In conclusion...



The product's saving grace was the new packaging, as its new formulation left users with a poor taste *on their lips*

Due to their cult following, they may not have a super hard time coming back from this as ratings and sentiments are already increasing

# Other Fun Wordclouds



Negative Title

n=4

'Cherry'





**Thank you!**