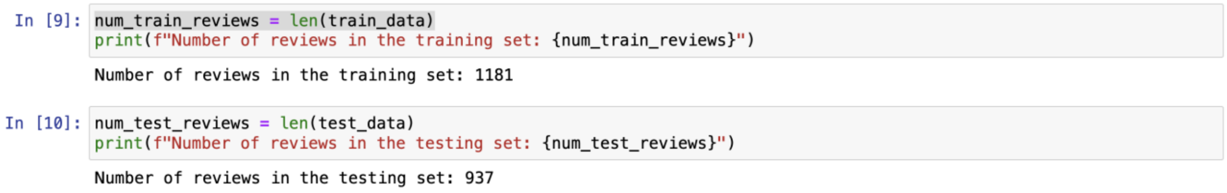
Assignment 2

Task 1: Number of reviews in the training set and the test set are 1181 and 937 respectively.

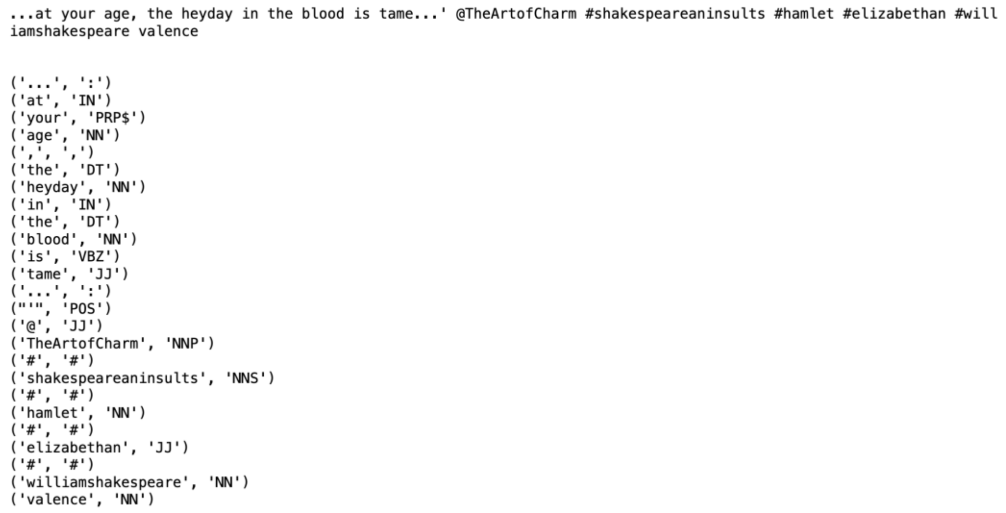


```
In [9]:  num_train_reviews = len(train_data)
          print(f"Number of reviews in the training set: {num_train_reviews}")

          Number of reviews in the training set: 1181

In [10]:  num_test_reviews = len(test_data)
          print(f"Number of reviews in the testing set: {num_test_reviews}")

          Number of reviews in the testing set: 937
```

Task 2: For sentiment analysis, the following parts of speech (POS) are particularly useful:

1. **Adjectives** (JJ, JJR, JJS): Adjectives are crucial because they describe the attributes of things or actions contributing directly to the sentiment conveyed in the text. For example, "happy", "sad", "angry" are all adjectives that help identify sentiment. In the first tweet, adjectives like "tame" in the context of a "heyday" might indicate a mild sentiment.
2. **Adverbs** (RB, RBR, RBS): Adverbs modify verbs or adjectives, providing context for the intensity or manner of an action. Words like "extremely", "barely", or "too" might help emphasizing the sentiment's intensity. In the second tweet, although no adverbs are highlighted in the provided tags, these could be useful for determining the intensity of emotional expressions.
3. **Verbs** (VB, VBD, VBG, VBN, VBP, VBZ): Verbs describe actions or states and can be directly tied to sentiment. Words like "love", "hate", "enjoy", or "deserve" express sentiments. For example, "hurt" in the second tweet, a verb, directly relates to negative sentiment.
4. **Nouns** (NN, NNS, NNP, NNPS): Nouns may indicate the subject or object of sentiment. In a tweet like "valence", which is a noun, the sentiment depends on the context and what the noun is related to. Words like "love", "success", and "failure" are often important in sentiment analysis as they represent key entities or outcomes that evoke sentiment.

Tweet 1:

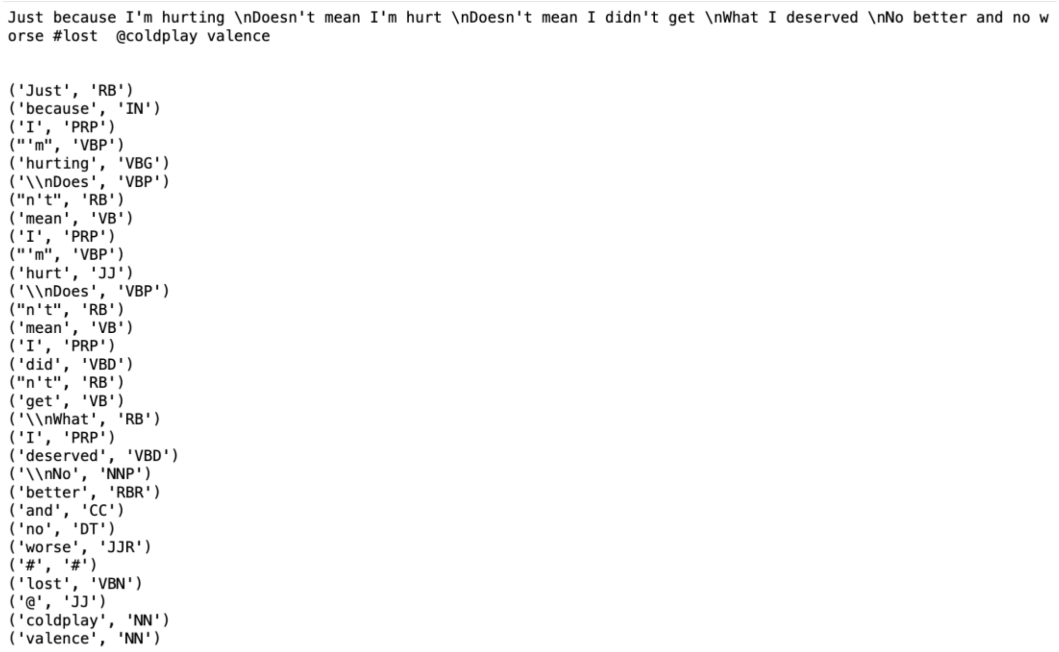


```
...at your age, the heyday in the blood is tame...' @TheArtofCharm #shakespeareaninsults #hamlet #elizabethan #will
iamshakespeare valence


('...', ':')
('at', 'IN')
('your', 'PRP$')
('age', 'NN')
(',', ',')
('the', 'DT')
('heyday', 'NN')
('in', 'IN')
('the', 'DT')
('blood', 'NN')
('is', 'VBZ')
('tame', 'JJ')
('...', ':')
("'", 'POS')
('@', 'JJ')
('TheArtofCharm', 'NNP')
('#', '#')
('shakespeareaninsults', 'NNS')
('#', '#')
('hamlet', 'NN')
('#', '#')
('elizabethan', 'JJ')
('#', '#')
('williamshakespeare', 'NN')
('valence', 'NN')
```

Key POS:

"tame" (JJ): This adjective conveys a sense of mildness or subdued emotion, which affects the sentiment. It suggests that the "heyday" may not be as exciting or intense as expected, which leans toward a neutral or slightly negative sentiment.

Tweet 2:



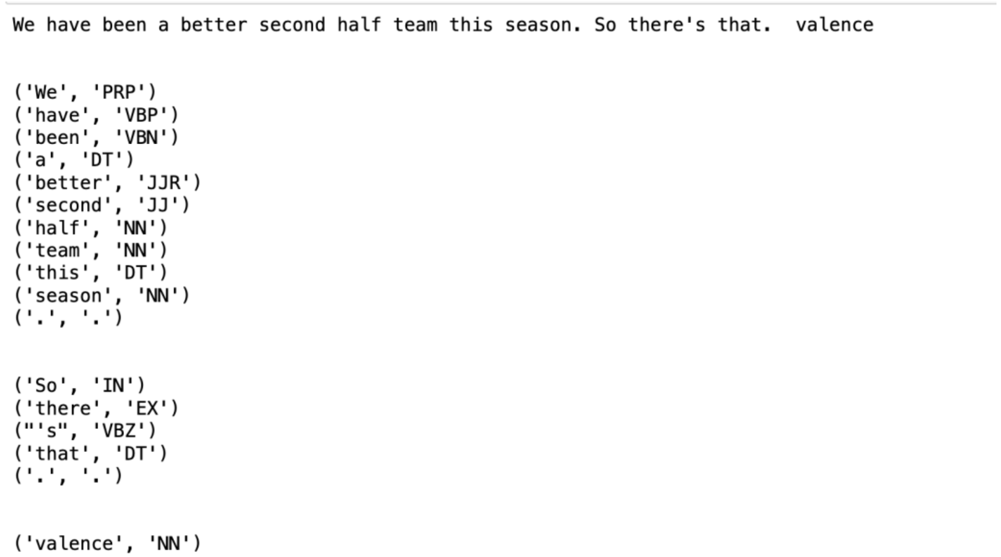
```
Just because I'm hurting \nDoesn't mean I'm hurt \nDoesn't mean I didn't get \nWhat I deserved \nNo better and no w
orse #lost  @coldplay valence


('Just', 'RB')
('because', 'IN')
('I', 'PRP')
("'m", 'VBP')
('hurting', 'VBG')
('\\nDoes', 'VBP')
("n't", 'RB')
('mean', 'VB')
('I', 'PRP')
("'m", 'VBP')
('hurt', 'JJ')
('\\nDoes', 'VBP')
("n't", 'RB')
('mean', 'VB')
('I', 'PRP')
('did', 'VBD')
("n't", 'RB')
('get', 'VB')
('\\nWhat', 'RB')
('I', 'PRP')
('deserved', 'VBD')
('\\nNo', 'NNP')
('better', 'RBR')
('and', 'CC')
('no', 'DT')
('worse', 'JJR')
('#', '#')
('lost', 'VBN')
('@', 'JJ')
('coldplay', 'NN')
('valence', 'NN')
```

Key POS:

The verbs "hurting" (VBG) and "hurt" (VBN) imply negative feelings, which adds to the tweet's overall negative tone.

The comparative adjectives "better" (JJR) and "worse" (JJR) characterize the difference between emotional states and add to the complexity of sentiment by expressing both progress and failure.

Tweet 3:



```
We have been a better second half team this season. So there's that.  valence


('We', 'PRP')
('have', 'VBP')
('been', 'VBN')
('a', 'DT')
('better', 'JJR')
('second', 'JJ')
('half', 'NN')
('team', 'NN')
('this', 'DT')
('season', 'NN')
('.', '.')


('So', 'IN')
('there', 'EX')
("'s", 'VBZ')
('that', 'DT')
('.', '.')


('valence', 'NN')
```

Key POS:

"better" (JJR): This adjective indicates improvement, which suggests a positive sentiment.
"valence" (NN): Like in Tweet 1, the use of "valence" again points to an overall assessment of sentiment.

Errors in POS Tagging

1. Errors with Contractions: In Tweet 2, "I'm" is split into "I" (PRP) and "'m" (VBP), which is fine in many cases but as discussed in class it might not capture the contraction correctly for sentiment analysis.
2. Misclassification of Hashtags: The hashtags (#lost, #shakespeareaninsults, etc.) are tagged as nouns (NN), which is technically correct. However, for sentiment analysis, they could carry additional meaning as a group of words and might need separate analysis.
3. Misclassification of "valence": The word "valence" is tagged as a noun (NN) in all three tweets, which is accurate, but it represents a more abstract concept of sentiment. Depending on context, its meaning can shift, and it might require contextual understanding to assess sentiment better.

Task 3:

Number of features in training set: 5039

Number of features in test set: 5039

Task 5:

Number of features in training set (Unigram + Bigram): 19390

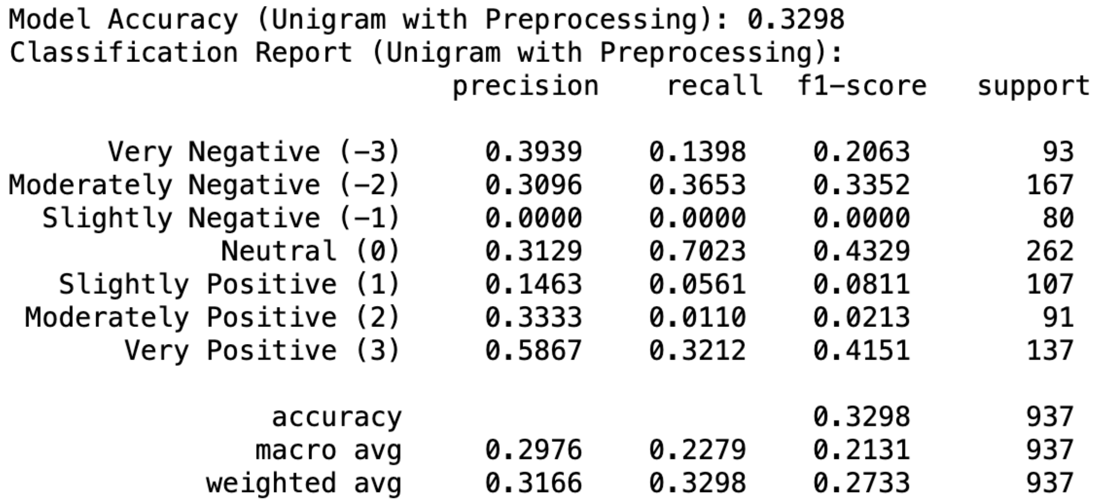
Number of features in test set (Unigram + Bigram): 19390

Task 6:

Number of features in training set (Unigram + Bigram + Trigram): 35185

Number of features in test set (Unigram + Bigram + Trigram): 35185

Task 8:



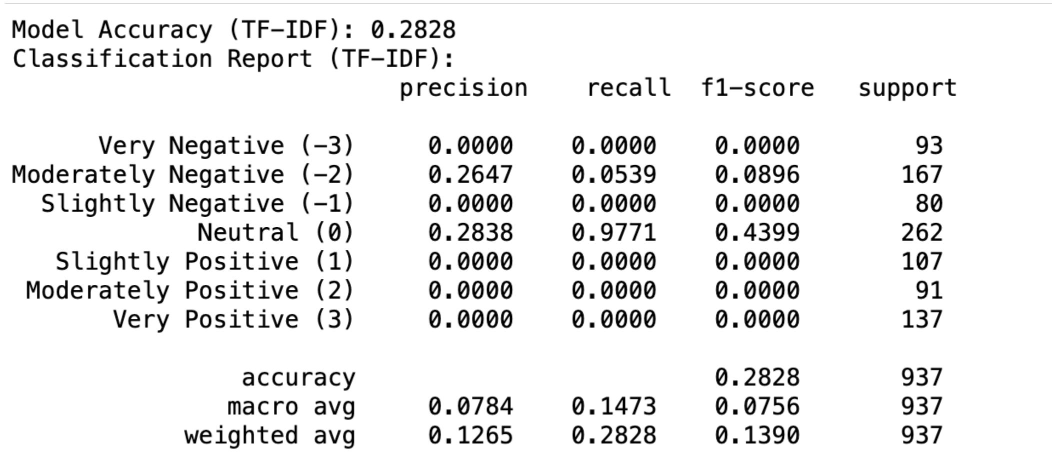
```
Model Accuracy (Unigram with Preprocessing): 0.3298
Classification Report (Unigram with Preprocessing):
                           precision    recall  f1-score   support

      Very Negative (-3)     0.3939    0.1398    0.2063        93
Moderately Negative (-2)     0.3096    0.3653    0.3352       167
  Slightly Negative (-1)     0.0000    0.0000    0.0000        80
             Neutral (0)     0.3129    0.7023    0.4329       262
   Slightly Positive (1)     0.1463    0.0561    0.0811       107
 Moderately Positive (2)     0.3333    0.0110    0.0213        91
       Very Positive (3)     0.5867    0.3212    0.4151       137

                accuracy                         0.3298       937
               macro avg     0.2976    0.2279    0.2131       937
            weighted avg     0.3166    0.3298    0.2733       937
```

Task 9:

Feature Set Analysis:

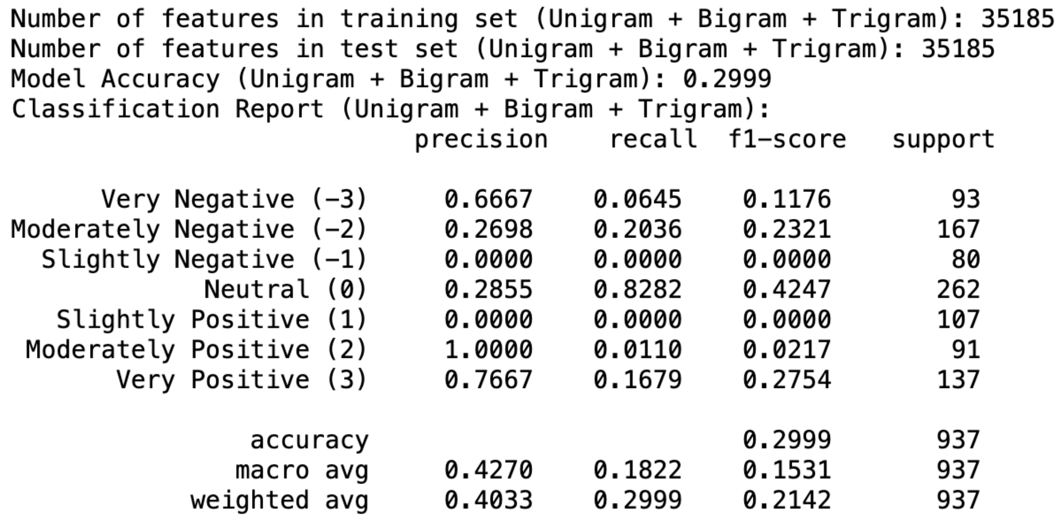
From the results, **unigram with preprocessing** combination gave the highest accuracy of **0.3298**. Here's a breakdown of how each feature set performed:

1. **TF-IDF (0.2828 accuracy)**:



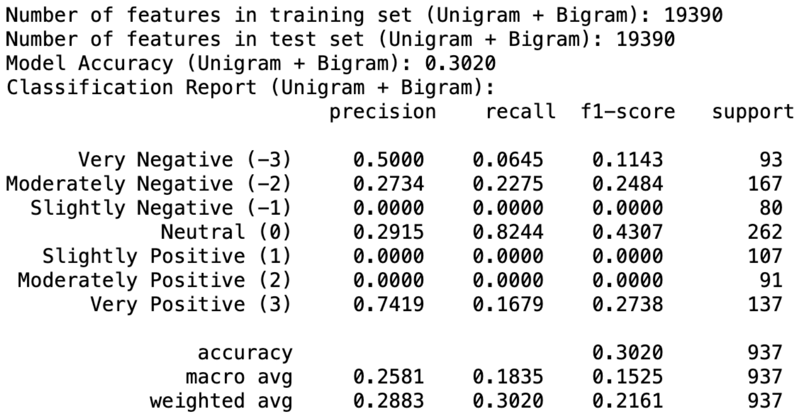
```
Model Accuracy (TF-IDF): 0.2828
Classification Report (TF-IDF):
                          precision    recall  f1-score   support

      Very Negative (-3)     0.0000    0.0000    0.0000        93
Moderately Negative (-2)     0.2647    0.0539    0.0896       167
  Slightly Negative (-1)     0.0000    0.0000    0.0000        80
             Neutral (0)     0.2838    0.9771    0.4399       262
   Slightly Positive (1)     0.0000    0.0000    0.0000       107
 Moderately Positive (2)     0.0000    0.0000    0.0000        91
       Very Positive (3)     0.0000    0.0000    0.0000       137

                accuracy                         0.2828       937
               macro avg     0.0784    0.1473    0.0756       937
            weighted avg     0.1265    0.2828    0.1390       937
```

2. **Unigram + Bigram + Trigram (0.2999 accuracy)**:



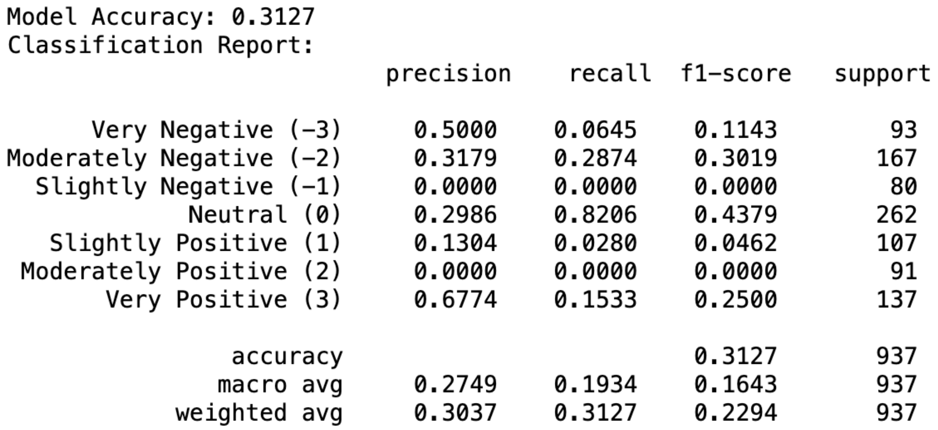
```
Number of features in training set (Unigram + Bigram + Trigram): 35185
Number of features in test set (Unigram + Bigram + Trigram): 35185
Model Accuracy (Unigram + Bigram + Trigram): 0.2999
Classification Report (Unigram + Bigram + Trigram):
                          precision    recall  f1-score   support

      Very Negative (-3)     0.6667    0.0645    0.1176        93
Moderately Negative (-2)     0.2698    0.2036    0.2321       167
  Slightly Negative (-1)     0.0000    0.0000    0.0000        80
             Neutral (0)     0.2855    0.8282    0.4247       262
   Slightly Positive (1)     0.0000    0.0000    0.0000       107
 Moderately Positive (2)     1.0000    0.0110    0.0217        91
       Very Positive (3)     0.7667    0.1679    0.2754       137

                accuracy                         0.2999       937
               macro avg     0.4270    0.1822    0.1531       937
            weighted avg     0.4033    0.2999    0.2142       937
```

3. **Unigram + Bigram (0.3020 accuracy)**:



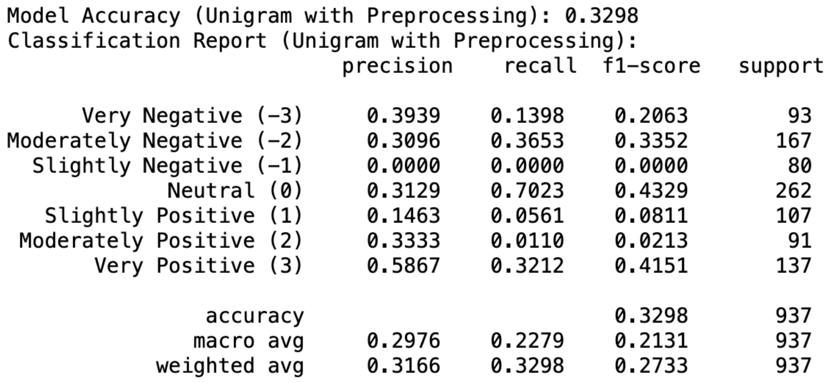
```
Number of features in training set (Unigram + Bigram): 19390
Number of features in test set (Unigram + Bigram): 19390
Model Accuracy (Unigram + Bigram): 0.3020
Classification Report (Unigram + Bigram):
                          precision    recall  f1-score   support

      Very Negative (-3)     0.5000    0.0645    0.1143        93
Moderately Negative (-2)     0.2734    0.2275    0.2484       167
  Slightly Negative (-1)     0.0000    0.0000    0.0000        80
             Neutral (0)     0.2915    0.8244    0.4307       262
   Slightly Positive (1)     0.0000    0.0000    0.0000       107
 Moderately Positive (2)     0.0000    0.0000    0.0000        91
       Very Positive (3)     0.7419    0.1679    0.2738       137

                accuracy                         0.3020       937
               macro avg     0.2581    0.1835    0.1525       937
            weighted avg     0.2883    0.3020    0.2161       937
```

4. **Unigram (0.3127 accuracy)**:



```
Model Accuracy: 0.3127
Classification Report:
                           precision    recall  f1-score   support

      Very Negative (-3)      0.5000    0.0645    0.1143        93
Moderately Negative (-2)      0.3179    0.2874    0.3019       167
  Slightly Negative (-1)      0.0000    0.0000    0.0000        80
             Neutral (0)      0.2986    0.8206    0.4379       262
     Slightly Positive (1)    0.1304    0.0280    0.0462       107
   Moderately Positive (2)    0.0000    0.0000    0.0000        91
       Very Positive (3)      0.6774    0.1533    0.2500       137

                accuracy                          0.3127       937
               macro avg      0.2749    0.1934    0.1643       937
            weighted avg      0.3037    0.3127    0.2294       937
```

5. **Unigram with Preprocessing (0.3298 accuracy)**:



```
Model Accuracy (Unigram with Preprocessing): 0.3298
Classification Report (Unigram with Preprocessing):
                           precision    recall  f1-score   support

      Very Negative (-3)      0.3939    0.1398    0.2063        93
Moderately Negative (-2)      0.3096    0.3653    0.3352       167
  Slightly Negative (-1)      0.0000    0.0000    0.0000        80
             Neutral (0)      0.3129    0.7023    0.4329       262
   Slightly Positive (1)      0.1463    0.0561    0.0811       107
 Moderately Positive (2)      0.3333    0.0110    0.0213        91
       Very Positive (3)      0.5867    0.3212    0.4151       137

                accuracy                          0.3298       937
               macro avg      0.2976    0.2279    0.2131       937
            weighted avg      0.3166    0.3298    0.2733       937
```

Why Unigram with Preprocessing Works Better:

Unigrams capture individual word importance without overfitting to higher-order word combinations that may be less representative of sentiment in short texts like tweets. Preprocessing helped eliminate irrelevant information, such as stop words or special characters, allowing the model to focus on the most informative words, which likely leads to more accurate sentiment predictions.

Performance by Label in Best-Performing Model (Unigram with Preprocessing)

Looking at the classification report for **unigram with preprocessing**, here's how the model performed for each sentiment label:

1. **Very Negative (-3)**: Precision = **0.3939**, Recall = **0.1398**, F1-score = **0.2063**

The model performs poorly for **Very Negative** examples, which might be due to their smaller presence in the dataset. There's a significant class imbalance, as the support for this label is only **93** examples.

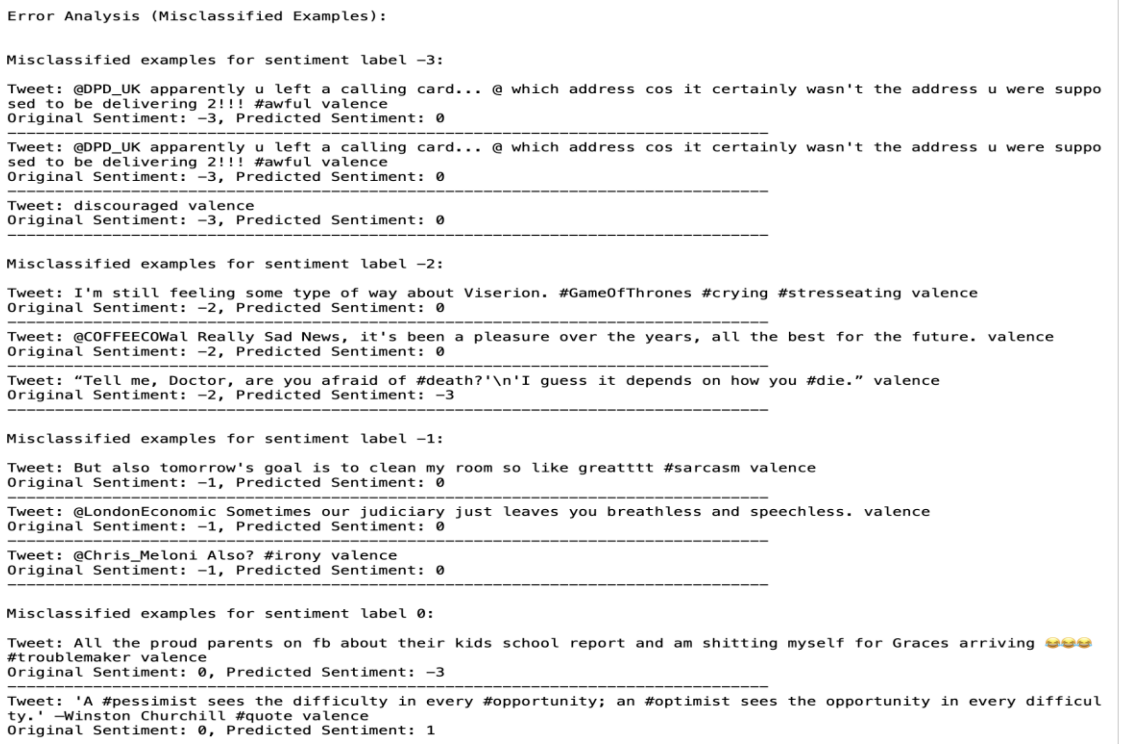
2. **Moderately Negative (-2)**: Precision = **0.3096**, Recall = **0.3653**, F1-score = **0.3352**
   This label has a better balance of precision and recall than the more extreme negative labels. It suggests that the model is somewhat effective at identifying moderately negative sentiment but still struggles with false positives.

3. **Slightly Negative (-1)**: Precision = **0.0000**, Recall = **0.0000**, F1-score = **0.0000**
   The model completely fails to identify **slightly negative** sentiments, which is likely due to the rarity of this class in the dataset or the difficulty in distinguishing it from neutral or moderately negative sentiments.
4. **Neutral (0)**: Precision = **0.3129**, Recall = **0.7023**, F1-score = **0.4329**
   Neutral sentiment has the highest recall, meaning the model can identify neutral examples quite well, but its precision is not as strong. This suggests that many neutral examples are being correctly classified, but some non-neutral tweets are also incorrectly labeled as neutral.
5. **Slightly Positive (1)**: Precision = **0.1463**, Recall = **0.0561**, F1-score = **0.0811**
   The performance for **slightly positive** sentiment is poor, likely due to a small number of examples in the dataset (107 tweets). This class might be underrepresented or too subtle for the model to capture effectively.
6. **Moderately Positive (2)**: Precision = **0.3333**, Recall = **0.0110**, F1-score = **0.0213**
   Moderately positive sentiment also performs poorly, especially in recall, where it has very few true positives. This is a common challenge with imbalanced datasets and highlights the difficulty of classifying moderately positive examples.
7. **Very Positive (3)**: Precision = **0.5867**, Recall = **0.3212**, F1-score = **0.4151**
   The model performs better for **very positive** sentiment compared to most other categories. The higher precision indicates fewer false positives, but recall is still not high, suggesting that some very positive examples are missed.

Inference and Recommendations:

- **Class Imbalance**: The low precision and recall for extreme labels (-3, -1, 1, 2, 3) suggest that the model is struggling with class imbalance. The model is biased toward labeling many instances as **neutral (0)**, as it has the highest recall.
- **Improvement Areas**:
  Resampling: To address the class imbalance, we can consider techniques such as oversampling the minority classes or under sampling the majority class (neutral).

Task 10:



```
Error Analysis (Misclassified Examples):

Misclassified examples for sentiment label -3:

Tweet: @DPD_UK apparently u left a calling card... @ which address cos it certainly wasn't the address u were suppo
sed to be delivering 2!!! #awful valence
Original Sentiment: -3, Predicted Sentiment: 0
--------------------------------------------------------------------------------
Tweet: @DPD_UK apparently u left a calling card... @ which address cos it certainly wasn't the address u were suppo
sed to be delivering 2!!! #awful valence
Original Sentiment: -3, Predicted Sentiment: 0
--------------------------------------------------------------------------------
Tweet: discouraged valence
Original Sentiment: -3, Predicted Sentiment: 0
--------------------------------------------------------------------------------

Misclassified examples for sentiment label -2:

Tweet: I'm still feeling some type of way about Viserion. #GameOfThrones #crying #stresseating valence
Original Sentiment: -2, Predicted Sentiment: 0
--------------------------------------------------------------------------------
Tweet: @COFFEECOWal Really Sad News, it's been a pleasure over the years, all the best for the future. valence
Original Sentiment: -2, Predicted Sentiment: 0
--------------------------------------------------------------------------------
Tweet: “Tell me, Doctor, are you afraid of #death?'\n'I guess it depends on how you #die.” valence
Original Sentiment: -2, Predicted Sentiment: -3
--------------------------------------------------------------------------------

Misclassified examples for sentiment label -1:

Tweet: But also tomorrow's goal is to clean my room so like greatttt #sarcasm valence
Original Sentiment: -1, Predicted Sentiment: 0
--------------------------------------------------------------------------------
Tweet: @LondonEconomic Sometimes our judiciary just leaves you breathless and speechless. valence
Original Sentiment: -1, Predicted Sentiment: 0
--------------------------------------------------------------------------------
Tweet: @Chris_Meloni Also? #irony valence
Original Sentiment: -1, Predicted Sentiment: 0
--------------------------------------------------------------------------------

Misclassified examples for sentiment label 0:

Tweet: All the proud parents on fb about their kids school report and am shitting myself for Graces arriving 😂😂😂
#troublemaker valence
Original Sentiment: 0, Predicted Sentiment: -3
--------------------------------------------------------------------------------
Tweet: 'A #pessimist sees the difficulty in every #opportunity; an #optimist sees the opportunity in every difficul
ty.' —Winston Churchill #quote valence
Original Sentiment: 0, Predicted Sentiment: 1
```

First, I thought it be because of #’s and @’s, but after removing them the accuracy dropped further, I found that it was because they contained words that provided insights to sentiments. Some of them are very surprising, and I genuinely have no clue to why the labels that lie very far away from each other in the spectrum are misclassified. One possible reason is that sarcasm and irony are difficult to detect with traditional text-based models.

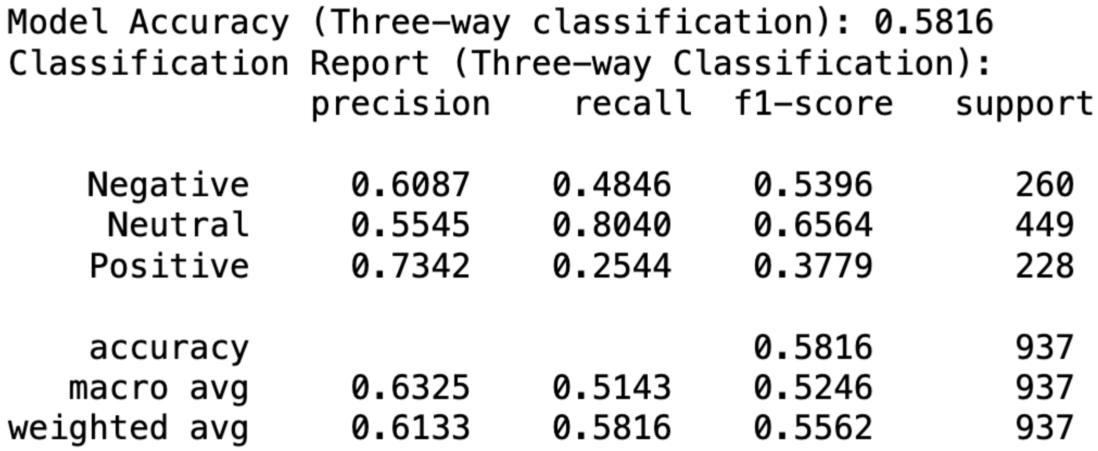
Some words can be interpreted differently based on tone, punctuation, or additional context. For example, "up early, kicking ass and taking names" could be seen as positive motivation or sarcastic frustration.

Possible improvements:-

1. Instead of removing hashtags and mentions, analyze them for sentiment-related trends. I implemented it but not in the best way.
2. Introduce additional training data specifically labeled for sarcasm and irony detection.
3. Class Balancing. In this dataset there was a lot of class-imbalance.

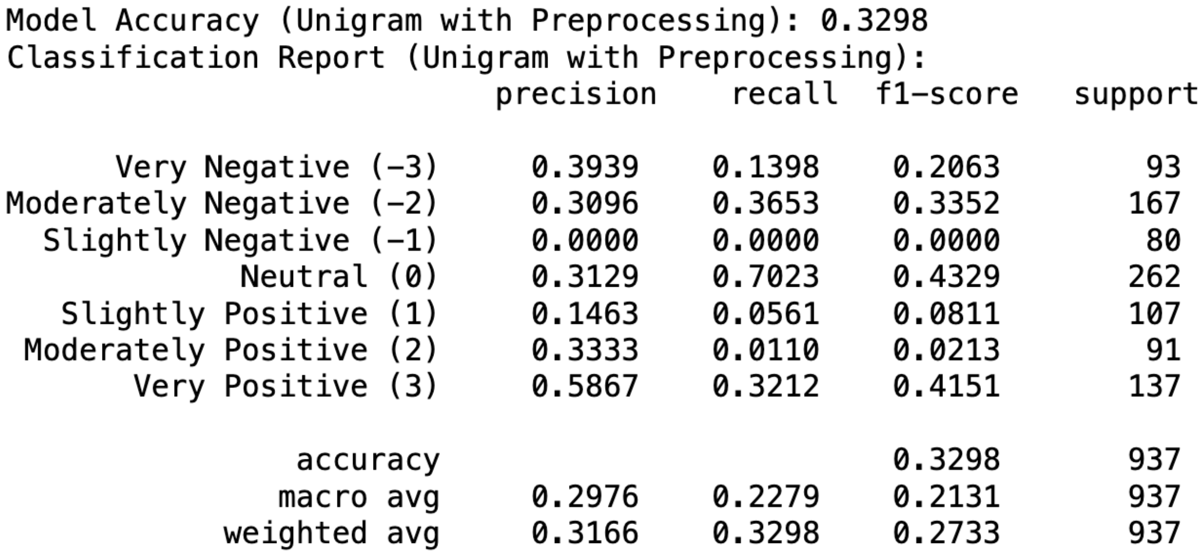
Task 11:

Three-way:



```
Model Accuracy (Three-way classification): 0.5816
Classification Report (Three-way Classification):
              precision    recall  f1-score   support

    Negative     0.6087    0.4846    0.5396       260
     Neutral     0.5545    0.8040    0.6564       449
    Positive     0.7342    0.2544    0.3779       228

    accuracy                         0.5816       937
   macro avg     0.6325    0.5143    0.5246       937
weighted avg     0.6133    0.5816    0.5562       937
```

Best 7-way:



```
Model Accuracy (Unigram with Preprocessing): 0.3298
Classification Report (Unigram with Preprocessing):
                         precision    recall  f1-score   support

     Very Negative (-3)     0.3939    0.1398    0.2063        93
Moderately Negative (-2)     0.3096    0.3653    0.3352       167
  Slightly Negative (-1)     0.0000    0.0000    0.0000        80
            Neutral (0)     0.3129    0.7023    0.4329       262
   Slightly Positive (1)     0.1463    0.0561    0.0811       107
 Moderately Positive (2)     0.3333    0.0110    0.0213        91
       Very Positive (3)     0.5867    0.3212    0.4151       137

               accuracy                         0.3298       937
              macro avg     0.2976    0.2279    0.2131       937
           weighted avg     0.3166    0.3298    0.2733       937
```

Since the three-way classification is a simpler task than the seven-way classification, the accuracy of the three-way classification is substantially higher. It is more difficult to discern between adjacent sentiment categories due to the seven-way classification's finer granularity. The model tends to default to neutrality when unsure, as indicated by the consistently high F1-score for the neutral label (0) across all models. Not only does the presence of an adjective matter, but so does the intensity of the sentiment. A sentence like "It was barely good" could be assigned a score of either 1 or 2, for instance, making it challenging for the model to appropriately represent such minute variations.