NLP Assignment 1

Evaluation Report

Team Members:
Akshat(2020172)
Arjun Mehra (2020178),
Anurag Gulati (2020176),
Aryman Srivastava (2020184)

1. Top 5 bigrams before smoothing and after each of the 2 selected smoothing techniques along with their probabilities BEFORE applying emotion component.

Top 5 Bigrams without Smoothing: [(('i', 'feel'), 0.02304450285507115), (('i', 'am'), 0.006639173388924137), (('feel', 'like'), 0.00559684582615789), (('i', 'was'), 0.004554518263391643), (('that', 'i'), 0.003716124354210097)]

Top 5 Bigrams with Laplace Smoothing: [(('i', 'feel'), 0.11043610327619874), (('i', 'am'), 0.03189412019960946), (('feel', 'like'), 0.0350976507217662), (('i', 'was'), 0.021913647211976566), (('that', 'i'), 0.02650602409638554)]

Top 5 Bigrams with Kneser-Ney Smoothing: [(('i', 'feel'), 0.26821417383616997), (('i', 'am'), 0.07713473335055371), (('feel', 'like'), 0.1504350549121223), (('i', 'was'), 0.05285392047116601), (('that', 'i'), 0.2050982733276795)]

When we compare the probabilities given for the top bigrams:

- "i feel" is more probable in Kneser-Ney smoothing than in Laplace smoothing.
- "i am", "feel like", "i was", and "that i" are also given higher probabilities with Kneser-Ney smoothing.

The higher probabilities in Kneser-Ney smoothing suggest that these bigrams are recognized as more contextually unique and less discounted than with Laplace smoothing. This is particularly important for bigrams involving common words ("i", "feel", "am", "like", "was", and "that"), where context plays a significant role in determining how likely they are to occur together.

Based on this comparison, Kneser-Ney smoothing is typically considered better for language modeling because it provides a more nuanced probability distribution that considers not just the frequency of the bigram itself but also the likelihood of the words within the bigram to occur in various contexts. This results in a more accurate model of language, which is why Kneser-Ney smoothing is often preferred in NLP applications over simpler methods like Laplace smoothing.

2. Reasoning for the method used for including emotion component

The method employed for integrating the emotion component into sentence generation utilizes a modified Bigram model. This model calculates the probabilities of subsequent sentences while incorporating emotion scores obtained for each sentence. Specifically, the emotion score corresponding to the desired emotion is added as a beta component to the probability calculation for each bigram, thereby influencing the likelihood of word selection based on emotional relevance.

For instance, when generating a sentence intended to convey anger, the anger emotion score is utilized for each bigram within the sentence. Adding the emotion score adjusts the overall probability of word candidacy, favoring words that are semantically suitable for the sentence context and highly associated with the target emotion. Consequently, words deemed highly suitable and strongly associated with the target emotion are assigned higher selection probabilities.

In cases where one component (either the suitability or the emotion score) is high while the other is low, the overall probability is still increased, albeit the normalization factor ensures that such instances do not unduly influence the outcome. Conversely, when the overall probability and emotion score are low for a particular word, the probability of selecting that word diminishes, effectively excluding it as a candidate for sentence generation. This mechanism ultimately facilitates the generation of sentences that align with the desired emotional tone by favoring emotionally relevant words within the given context.

Furthermore, the incorporation of emotion scores at the bigram level ensures consistency, as the emotion-influenced probabilities are applied uniformly across all potential word pairs within the sentence. This approach effectively treats the selection of bigrams as representative of the entire sentence, ensuring coherence in the emotional tone throughout the generated text.

3. 2 generated samples for each emotion, for a total of 12 generated samples

Anger

- akin sun discontent farm marriages racked costume
- accusing doomed worried cherish touristy younger simpler whose gathering day

Love

- history christmas dull youthful weight blinds hot
- share frightened kisses constructions lipsticks devoted bullshit

Sadness

- stopping paralysed zealand myself rarely major losing naman clench

- indulge killed nap divorce tons disregarded impoverished tragedy cramps rainbows

Surprise
- grade prospects shocked impacted had local princess
- garden waited vitamins taste stunned brimming scared upsetting

Fear
- unusually fateh arena move cramps hsps cypress
- cos freakishly slalom despair sunset touched interacting hart finance

Joy
- floor chapters yummy easier toronto significant literature
- rylin presenting back smell interested handle strayed out crazy melancholy

4. Accuracy and macro F1 scores obtained from extrinsic evaluation

```
Best Parameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
Test Accuracy: 0.7766666666666667
Classification Report:
              precision    recall  f1-score   support

       anger       0.79      0.73      0.76        50
        fear       0.75      0.75      0.75        50
         joy       0.66      0.73      0.69        50
        love       0.76      0.84      0.80        50
     sadness       0.73      0.65      0.69        50
    surprise       0.94      0.90      0.92        50

    accuracy                           0.77       300
   macro avg       0.77      0.77      0.77       300
weighted avg       0.77      0.77      0.77       300
```

5. For each emotion, pick 1 of the generated sample and the reason why it is generated according to its corresponding emotion.

Anger
- akin sun discontent farm marriages racked costume
- Words such as "discontent," "racked," and "costume" carry negative connotations. "Discontent" implies dissatisfaction or unhappiness, "racked" suggests torment or distress, and "costume" may connote artificiality or pretense. These negative associations can elicit feelings of anger or resentment.

Love
- history christmas dull youthful weight blinds hot
- The juxtaposition of contrasting elements, such as "dull" and "hot," "weight" and "youthful," creates a sense of tension or conflict. However, this tension can be interpreted as symbolic of the complexities of love. Love often involves navigating through contrasts and resolving differences, leading to a deeper connection and appreciation.

Sadness
- stopping paralysed zealand myself rarely major losing naman clench
- The inclusion of "myself" suggests a focus on individual experiences or internal struggles. This self-referential aspect, coupled with words like "stopping", “paralysed”, “losing” and "rarely" conveys a sense of isolation, introspection or negative meaning, which can be associated with feelings of loneliness or sadness.

Surprise
- grade prospects shocked impacted had local princess
- The juxtaposition of seemingly unrelated words such as "grade," "prospects," "local," and "princess" creates cognitive dissonance. These word combinations are unexpected and unconventional, leading the reader to pause and reevaluate the sentence for coherence.

Fear
- unusually fateh arena move cramps hsps cypress
- The use of words like "unusually," "cramps," and "hsps" conveys a sense of discomfort and unease. "Unusually" implies something out of the ordinary or unexpected, while "cramps" suggests physical discomfort or pain. The unfamiliar acronym "hsps" adds to the sense of disconcertion by introducing an element of mystery or confusion.

Joy
- floor chapters yummy easier toronto significant literature
- Many of the words in the sentence have positive connotations. "Yummy" conveys a sense of deliciousness and enjoyment, "significant" implies importance or value, and

“easier” implies easy work. "joy" is often associated with positive emotions, such as happiness and contentment.

6. A credit statement reflecting the contribution of each member of the group for the assignment.

Name	Main Contributions
Akshat (2020172)	Task 2 (1st), Compilation of codes
Anurag (2020176)	Task 2 (2nd subpart), Code optimization
Arjun (2020178)	Task 1, Task 2 (2nd subpart)
Aryman (2020184)	Task 2 (3rd and 4th subparts), Compilation of codes