NLP-3

Narasimha Rohit Katta

February 2024

1. (a)

Naive Bayesian Classifier Probabilities

Given the data, the following probabilities are computed:

Prior Probabilities:

$$P(Y) = \frac{\text{Number of 'Y' labels}}{\text{Total number of documents}} = \frac{5}{9}$$

$$P(N) = \frac{\text{Number of 'N' labels}}{\text{Total number of documents}} = \frac{4}{9}$$

Likelihoods for 'Y':

$$P(\text{blue}|Y) = \frac{\text{Number of times 'blue' appears with 'Y'}}{\text{Total number of words in 'Y' documents}} = \frac{2}{10}$$

$$P(\text{cautious}|Y) = \frac{2}{10}$$

$$P(\text{new}|Y) = \frac{1}{10}$$

$$P(\text{gloves}|Y) = \frac{1}{10}$$

$$P(\text{hat}|Y) = \frac{2}{10}$$

$$P(\text{cat}|Y) = \frac{2}{10}$$

Likelihoods for 'N':

$$P(\text{cautious}|N) = \frac{2}{8}$$
$$P(\text{new}|N) = \frac{2}{8}$$
$$P(\text{gloves}|N) = \frac{1}{8}$$
$$P(\text{hat}|N) = \frac{1}{8}$$
$$P(\text{cat}|N) = \frac{2}{8}$$

1. (b)

Classification of the New Document "Cautious Gloves"

Using the Naive Bayesian classifier probabilities computed earlier, we can classify the new document "cautious gloves" as follows:

For label 'Y':

$$P(Y|\text{"cautious gloves"}) \propto P(\text{cautious}|Y) \times P(\text{gloves}|Y) \times P(Y)$$
$$P(Y|\text{"cautious gloves"}) \propto \frac{2}{10} \times \frac{1}{10} \times \frac{5}{9}$$
$$P(Y|\text{"cautious gloves"}) \propto \frac{10}{900}$$
$$P(Y|\text{"cautious gloves"}) \propto \frac{1}{90}$$

For label 'N':

$$P(N|\text{"cautious gloves"}) \propto P(\text{cautious}|N) \times P(\text{gloves}|N) \times P(N)$$
$$P(N|\text{"cautious gloves"}) \propto \frac{2}{8} \times \frac{1}{8} \times \frac{4}{9}$$
$$P(N|\text{"cautious gloves"}) \propto \frac{8}{576}$$
$$P(N|\text{"cautious gloves"}) \propto \frac{1}{72}$$

Since $\frac{1}{72} > \frac{1}{90}$, the label 'N' has a higher posterior probability. Therefore, we classify the new document "cautious gloves" as 'N'.

2.Penn Treebank Tagging

Sentence (a):
I - PRP, take - VBP, steps - NNS, forward - RB, every - DT, day - NN, no - DT, matter - NN, how - WRB, small - JJ, they - PRP, may - MD, be - VB

Sentence (b):
I - PRP, solemnly - RB, swear - VBP, that - IN, I - PRP, am - VBP, up - RP, to - TO, no - DT, good - JJ

Sentence (c):
Show - VB, what - WP, we - PRP, truly - RB, are - VBP

Sentence (d):
The - DT, pen - NN, is - VBZ, mightier - JJR, than - IN, the - DT, sword - NN

Sentence (e):
Live - VB, in - IN, the - DT, now - NN

Challenges:

- Contextual meanings: Words can have different tags based on the context.
- Fixed phrases and idioms: These may not follow standard tagging rules.
- Ambiguity: Some words can be tagged in multiple ways depending on interpretation.

3. (a)

Calculating the Probability of the Sentence ”Patrick can see Cherry”

Given the Hidden Markov Model with initial probabilities, transition probabilities, and emission probabilities, we want to compute the probability of the sentence ”Patrick can see Cherry”. Assuming the most likely sequence of parts of speech is Noun (N) -¿ Modal (M) -¿ Verb (V) -¿ Noun (N), we calculate as follows:

Initial State Probability:

$$P(N) = 0.7$$

Emission Probabilities:

$$P(\text{”Patrick”—N}) = 0.3$$

$$P(\text{”can”—M}) = 0.4$$

$$P(\text{”see”—V}) = 0.5$$

$$P(\text{”Cherry”—N}) = 0.2$$

Transition Probabilities:

$$P(M|N) = 0.3$$
$$P(V|M) = 0.5$$
$$P(N|V) = 0.8 \quad \text{(Corrected value)}$$

Probability of the Sentence:

The probability of the sentence is the product of the initial state probability, the emission probabilities of the words given their states, and the transition probabilities between the states.

$$P(\text{Sentence}) = P(N) \times P(\text{”Patrick”—N}) \times P(M|N) \times P(\text{”can”—M}) \times P(V|M) \times P(\text{”see”—V}) \times P(N|V) \times P(\text{”Che}$$

$$P(\text{Sentence}) = 0.7 \times 0.3 \times 0.3 \times 0.4 \times 0.5 \times 0.5 \times 0.8 \times 0.2$$
$$P(\text{Sentence}) = 0.001008$$

Thus, the sentence ”Patrick can see Cherry” has a probability of 0.001008 of occurring according to the given Hidden Markov Model.

sub-question (b):

Probability Calculation for ”will Cherry spot Patrick”

Given the Hidden Markov Model (HMM) with initial probabilities, transition probabilities, and emission probabilities, we want to determine the likelihood of the sentence ”will Cherry spot Patrick” occurring.

Model Parameters

Let’s denote:

$$M \text{ for Modal}$$
$$N \text{ for Noun}$$
$$V \text{ for Verb}$$

The HMM provides the following probabilities:

$P(M) = 0.1$ (Initial probability for Modal)
$P($”will”—M$) = 0.6$ (Emission probability of ”will” given Modal)
$P(N|M) = 0.4$ (Transition probability from Modal to Noun)
$P($”Cherry”—N$) = 0.2$ (Emission probability of ”Cherry” given Noun)
$P(V|N) = 0.5$ (Transition probability from Noun to Verb)
$P($”spot”—V$) = 0.2$ (Emission probability of ”spot” given Verb)
$P(N|V) = 0.8$ (Transition probability from Verb to Noun)
$P($”Patrick”—N$) = 0.3$ (Emission probability of ”Patrick” given Noun)

Probability Calculation

The probability of the sentence is the product of these probabilities:

$$P(\text{Sentence}) = 0.1 \times 0.6 \times 0.4 \times 0.2 \times 0.5 \times 0.2 \times 0.8 \times 0.3 = 0.0001152$$

Therefore, the probability of the sentence ”will Cherry spot Patrick” occurring is 0.0001152.

sub-question (c):

Calculating the Most Likely Tag Sequence for ”Patrick can see Cherry”

Given the Hidden Markov Model with states $S = \{N, M, V\}$ and observations $K = \{\text{”Patrick”}, \text{”can”}, \text{”see”}, \text{”Cherry”}\}$, we calculate the most likely tag sequence for the sentence ”Patrick can see Cherry”. We use the initial probabilities, transition probabilities, and emission probabilities as provided.

Step 1: Initialization for ”Patrick”

$$\begin{aligned}
P(N|\text{"Patrick"}) &= \pi(N) \times P(\text{"Patrick"}|N) \\
&= 0.7 \times 0.3 \\
&= 0.21 \\
P(M|\text{"Patrick"}) &= \pi(M) \times P(\text{"Patrick"}|M) \\
&= 0.1 \times 0 \\
&= 0 \\
P(V|\text{"Patrick"}) &= \pi(V) \times P(\text{"Patrick"}|V) \\
&= 0.2 \times 0 \\
&= 0
\end{aligned}$$

The most likely tag for ”Patrick” is N (Noun).

Step 2: Transition for ”can”

For M (Modifier) following N:

$$\begin{aligned}
P(M|\text{"can"}, \text{previous} = N) &= P(N) \times P(N \rightarrow M) \times P(\text{"can"}|M) \\
&= 0.21 \times 0.3 \times 0.4 \\
&= 0.0252
\end{aligned}$$

The most likely tag for ”can” is M (Modifier).

Step 3: Transition for ”see”

For V (Verb) following M:

$$\begin{aligned}
P(V|\text{"see"}, \text{previous} = M) &= P(M) \times P(M \rightarrow V) \times P(\text{"see"}|V) \\
&= 0.0252 \times 0.5 \times 0.5 \\
&= 0.0063
\end{aligned}$$

The most likely tag for ”see” is V (Verb).

Step 4: Transition for ”Cherry”

For N (Noun) following V:

$$\begin{aligned}
P(N|\text{"Cherry"}, \text{previous} = V) &= P(V) \times P(V \rightarrow N) \times P(\text{"Cherry"}|N) \\
&= 0.0063 \times 0.8 \times 0.2 \\
&= 0.001008
\end{aligned}$$

The most likely tag for ”Cherry” is N (Noun).

Therefore, the most likely sequence of tags for the sentence ”Patrick can see Cherry” is NMVN.

sub-question (d):

Most Likely Tag Sequence for ”will Cherry spot Patrick”

1. **Initialization for ”will”:**
   - For M (Modifier):

$$P(\text{M}|\text{”will”}) = \pi(\text{M}) \times P(\text{”will”}|\text{M}) = 0.1 \times 0.6 = 0.06$$

2. **Transition from ”will” to ”Cherry”:**
   - For N (Noun) following M (Modifier):

$$P(\text{N}|\text{”Cherry”}) = P(\text{M})\times P(\text{M} \rightarrow \text{N})\times P(\text{”Cherry”}|\text{N}) = 0.06\times 0.4\times 0.2 = 0.0048$$

3. **Transition from ”Cherry” to ”spot”:**
   - For V (Verb) following N (Noun):

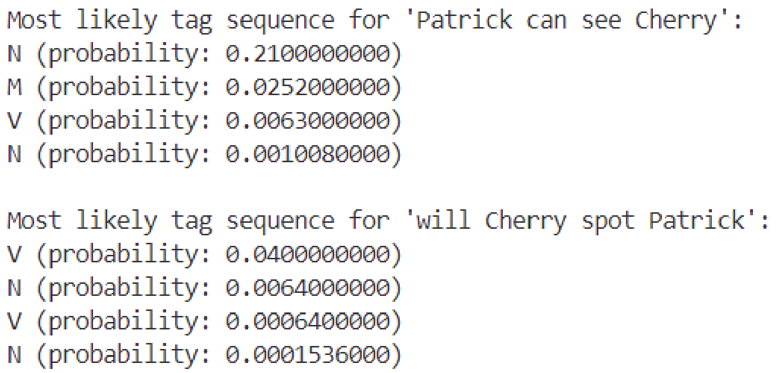
$$P(\text{V}|\text{”spot”}) = P(\text{N})\times P(\text{N} \rightarrow \text{V})\times P(\text{”spot”}|\text{V}) = 0.0048\times 0.5\times 0.2 = 0.00048$$

4. **Transition from ”spot” to ”Patrick”:**
   - For N (Noun) following V (Verb):

$$P(\text{N}|\text{”Patrick”}) = P(\text{V})\times P(\text{V} \rightarrow \text{N})\times P(\text{”Patrick”}|\text{N}) = 0.00048\times 0.8\times 0.3 = 0.0001152$$

According to these calculations, the most likely tag sequence for the sentence ”will Cherry spot Patrick” is M-N-V-N.

sub-question (e):



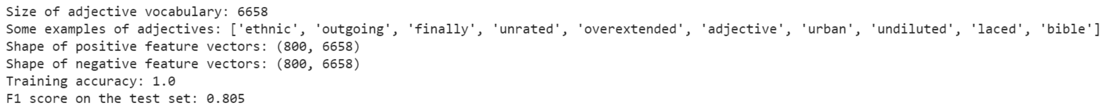
```
Most likely tag sequence for 'Patrick can see Cherry':
N (probability: 0.2100000000)
M (probability: 0.0252000000)
V (probability: 0.0063000000)
N (probability: 0.0010080000)

Most likely tag sequence for 'will Cherry spot Patrick':
V (probability: 0.0400000000)
N (probability: 0.0064000000)
V (probability: 0.0006400000)
N (probability: 0.0001536000)
```

sub-question (f):

The most likely sequences obtained by running the code may differ from the sequences obtained in parts (c) and (d). In the example of ”will Cherry spot Patrick,” the Viterbi algorithm identified ”V-N-V-N” as the most probable sequence, whereas the initial assumption based on linguistic intuition was ”M-N-V-N.”

If there is a discrepancy between the sequences, it is likely due to the algorithm’s reliance on the HMM parameters rather than on linguistic intuition. The Viterbi algorithm calculates the most likely sequence based on the probabilities defined in the model, which can sometimes lead to unexpected results that deviate from traditional linguistic assumptions. This demonstrates the data-driven nature of the Viterbi algorithm and its ability to uncover patterns based on the probabilities specified in the HMM.

4. **sub-question (a-f):**



```
Size of adjective vocabulary: 6658
Some examples of adjectives: ['ethnic', 'outgoing', 'finally', 'unrated', 'overextended', 'adjective', 'urban', 'undiluted', 'laced', 'bible']
Shape of positive feature vectors: (800, 6658)
Shape of negative feature vectors: (800, 6658)
Training accuracy: 1.0
F1 score on the test set: 0.805
```

sub-question (g):

The results demonstrate that the features based on adjectives were somewhat effective for the classification task:

1. Training Accuracy: The model achieved a training accuracy of 100

2. F1 Score on the Test Set: The F1 score of 0.805 on the test set indicates that the model performed well but not flawlessly. The discrepancy between the training accuracy and the F1 score suggests potential overfitting.

Overall, the adjective-based features seem effective but with limitations:

- Overfitting: The perfect training accuracy hints at overfitting. Employing regularization techniques or increasing the training data size could mitigate this issue.

- Feature Limitations: Adjectives capture sentiment but may miss other textual nuances. Including other parts of speech, such as adverbs and verbs, could enhance the feature set.

- Additional Features: Incorporating bigrams or trigrams as features could improve model performance by capturing more contextual information.

In conclusion, while the adjective-based features were effective to a certain extent, there is room for improvement by addressing overfitting and expanding the feature set.