In Class Assignemnt 7

October 30, 2024

1 In-Class Activity: Building and Plotting a Language Model

Aaron Vo
CS 584
October 30, 2024

1.1 Objective

The goal of this activity is to build a simple language model by tokenizing a text corpus, counting word occurrences, and plotting the frequency distribution of words.

1.2 Instructions

1. Tokenize the Corpus: Use the following corpus for tokenization. "The quick brown fox jumps over the lazy dog. The fox was quick to jump." Tokenize the text into individual words. Be sure to: • Exclude punctuation. • Convert all words to lowercase.
2. Build a Frequency Dictionary: After tokenizing, count the occur- rences of each unique word and create a dictionary that maps each word to its frequency. The result should look like this: {"the": 3, "quick": 2, "fox": 2, ...}
3. Convert to a Probability Distribution: Calculate the probability of each word by dividing its frequency by the total number of words. This will create a simple unigram language model: P (word) = Frequency of word Total number of words
4. Plot the Distribution: Use Python's matplotlib library to create a bar plot of the probability distribution. The x-axis should

[18]:
```
import re
from collections import defaultdict
import matplotlib.pyplot as plt
```

Step 1: Tokenize

[6]:
```
text = "The quick brown fox jumps over the lazy dog. The fox was quick to jump."
tokens = re.findall(r'\b\w+\b', text.lower())
print(tokens)
```

```
['the', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog', 'the',
'fox', 'was', 'quick', 'to', 'jump']
```

Step 2: Build the dictionary

[7]:
```
frequency_dict = defaultdict(int)
for token in tokens:
    frequency_dict[token] += 1
```

[9]:
```
print(frequency_dict)
```

```
defaultdict(<class 'int'>, {'the': 3, 'quick': 2, 'brown': 1, 'fox': 2, 'jumps':
1, 'over': 1, 'lazy': 1, 'dog': 1, 'was': 1, 'to': 1, 'jump': 1})
```

Step 4: Convert to a Probability Distribution

[11]:
```
print(len(tokens))
```

```
15
```

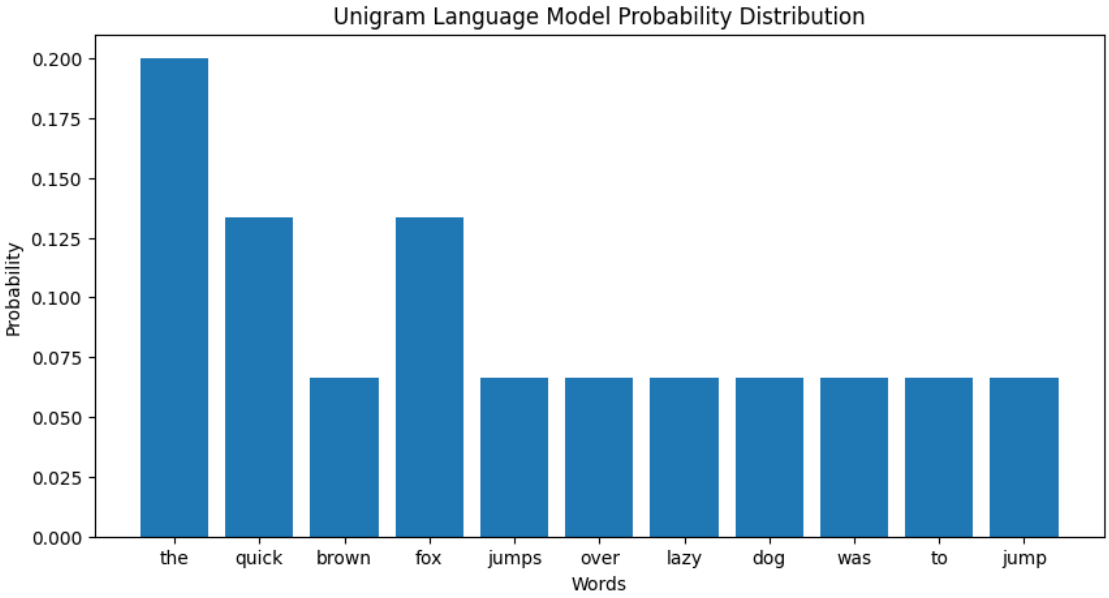
[14]:
```
prob_dist = defaultdict(float)
for key in frequency_dict.keys():
    prob_dist[key] = frequency_dict[key]/len(tokens)
```

[15]:
```
print(prob_dist)
```

```
defaultdict(<class 'float'>, {'the': 0.2, 'quick': 0.13333333333333333, 'brown':
0.06666666666666667, 'fox': 0.13333333333333333, 'jumps': 0.06666666666666667,
'over': 0.06666666666666667, 'lazy': 0.06666666666666667, 'dog':
0.06666666666666667, 'was': 0.06666666666666667, 'to': 0.06666666666666667,
'jump': 0.06666666666666667})
```

Step 5: Plot the Distribution

[20]:
```
plt.figure(figsize=(10, 5))
plt.bar(prob_dist.keys(), prob_dist.values())
plt.xlabel("Words")
plt.ylabel("Probability")
plt.title("Unigram Language Model Probability Distribution")
plt.show()
```


Unigram Language Model Probability Distribution
Probability
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
the
quick
brown
fox
jumps
over
lazy
dog
was
to
jump
Words