In [53]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [55]:
```
df = pd.read_csv(r"C:\Users\Vaish\Downloads\datasets-main\datasets-main\tripadvisor
df.head()
```

Out[55]:

	Review	Rating
0	nice hotel expensive parking got good deal sta...	4
1	ok nothing special charge diamond member hilto...	2
2	nice rooms not 4* experience hotel monaco seat...	3
3	unique, great stay, wonderful time hotel monac...	5
4	great stay great stay, went seahawk game aweso...	5

In [57]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20491 entries, 0 to 20490
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Review  20491 non-null  object
 1   Rating  20491 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 320.3+ KB
```

In [59]:
```
import re
from nltk.corpus import stopwords
def clean(review):
    review = review.lower()
    review = re.sub('[^a-z A-Z 0-9-]+', '', review)

    review = " ".join([word for word in review.split() if word not in stopwords.wor

    return review
```

In [61]:
```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Vaish\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[61]: True

In [63]:
```
def clean(text):

    import re
```

```
    text = re.sub(r'[^\w\s]', '', text)
    return text.lower()
```

In [65]:
```
def corpus(text):
    return text.split()
```

In [67]:
```
df['Review_lists'] = df['Review'].apply(corpus)
```

In [69]:
```
from tqdm import trange

corpus = []

for i in trange(df.shape[0], ncols=100, colour='green', smoothing=0.8):
    corpus += df['Review_lists'][i]
```

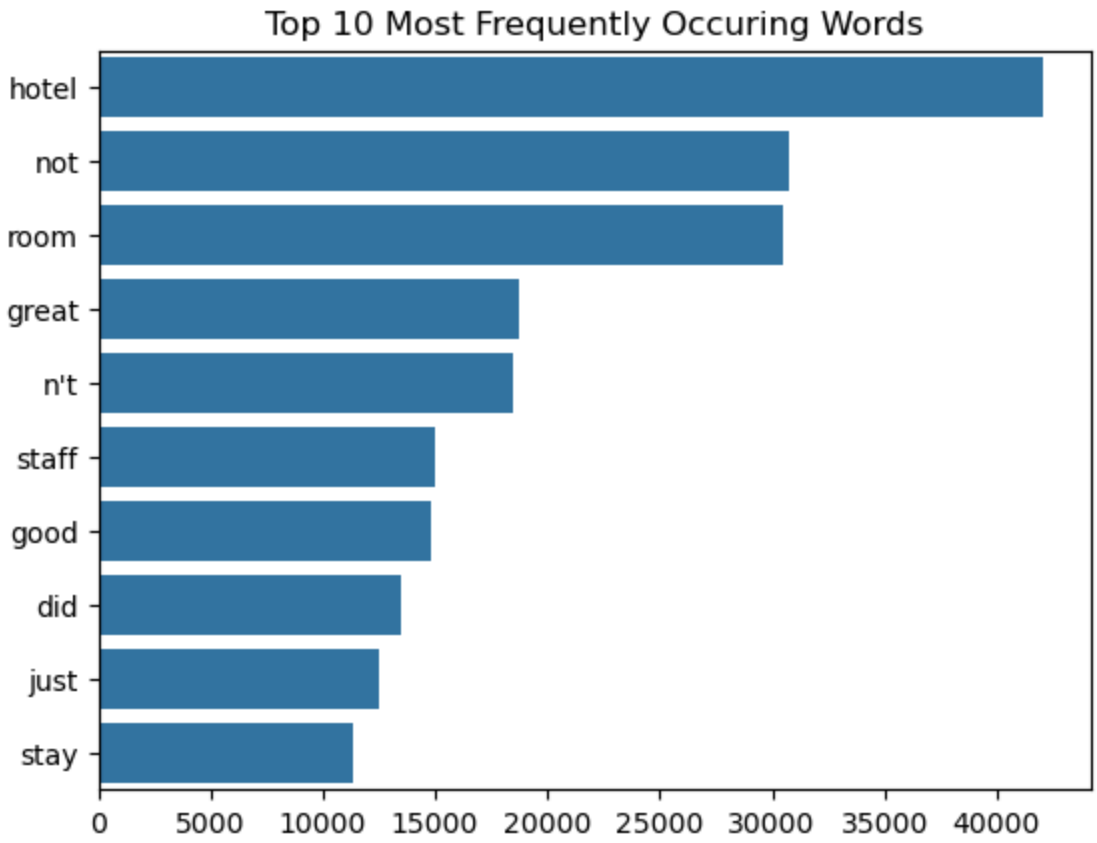
```
100%|████████████████████████████████████████████████████| 20491/20491 [00:00<00:0
0, 77750.36it/s]
```

In [71]:
```
from collections import Counter

mostCommon = Counter(corpus).most_common(10)  # Top 10 most common words
print(mostCommon)
```

```
[('hotel', 42079), ('not', 30750), ('room', 30532), ('great', 18732), ("n't", 1843
6), ('staff', 14950), ('good', 14791), ('did', 13433), ('just', 12458), ('stay', 113
76)]
```

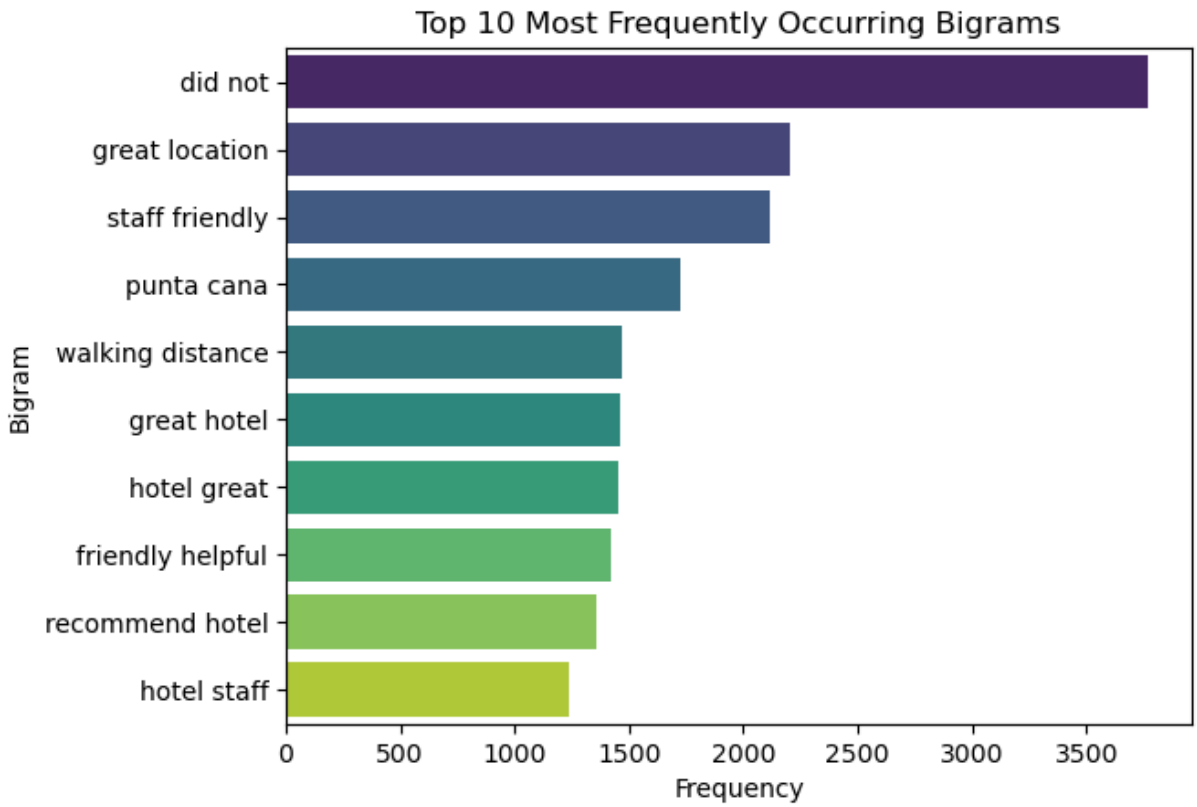
In [73]:
```
words = []

freq = []

for word, count in mostCommon:

    words.append(word)

    freq.append(count)
```

In [75]:
```
import seaborn as sns

sns.barplot(x=freq, y=words)

plt.title('Top 10 Most Frequently Occuring Words')

plt.show()
```



In [87]:
```
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
cv = CountVectorizer(ngram_range=(2, 2), max_features=5000, dtype=np.int32)

bigrams = cv.fit_transform(df['Review'])
count_values = bigrams.sum(axis=0).A1
ngram_freq = pd.DataFrame(sorted([(count_values[i], k) for k, i in cv.vocabulary_.i
                                  reverse=True),
                           columns=["frequency", "ngram"])
sns.barplot(x=ngram_freq['frequency'][:10], y=ngram_freq['ngram'][:10], palette="vi
plt.title('Top 10 Most Frequently Occurring Bigrams')
plt.xlabel('Frequency')
plt.ylabel('Bigram')
plt.show()
```

```
C:\Users\Vaish\AppData\Local\Temp\ipykernel_20184\1184248313.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=ngram_freq['frequency'][:10], y=ngram_freq['ngram'][:10], palette="v
iridis")
```


Top 10 Most Frequently Occurring Bigrams
Bigram
did not
great location
staff friendly
punta cana
walking distance
great hotel
hotel great
friendly helpful
recommend hotel
hotel staff
0
500
1000
1500
2000
2500
3000
3500
Frequency

In [ ]:

In [ ]: