

Social Computing (CS60017) Assignment 2

Aspect based Sentiment Classification from Reviews on Social Media Sites

Name : Amit Kumar

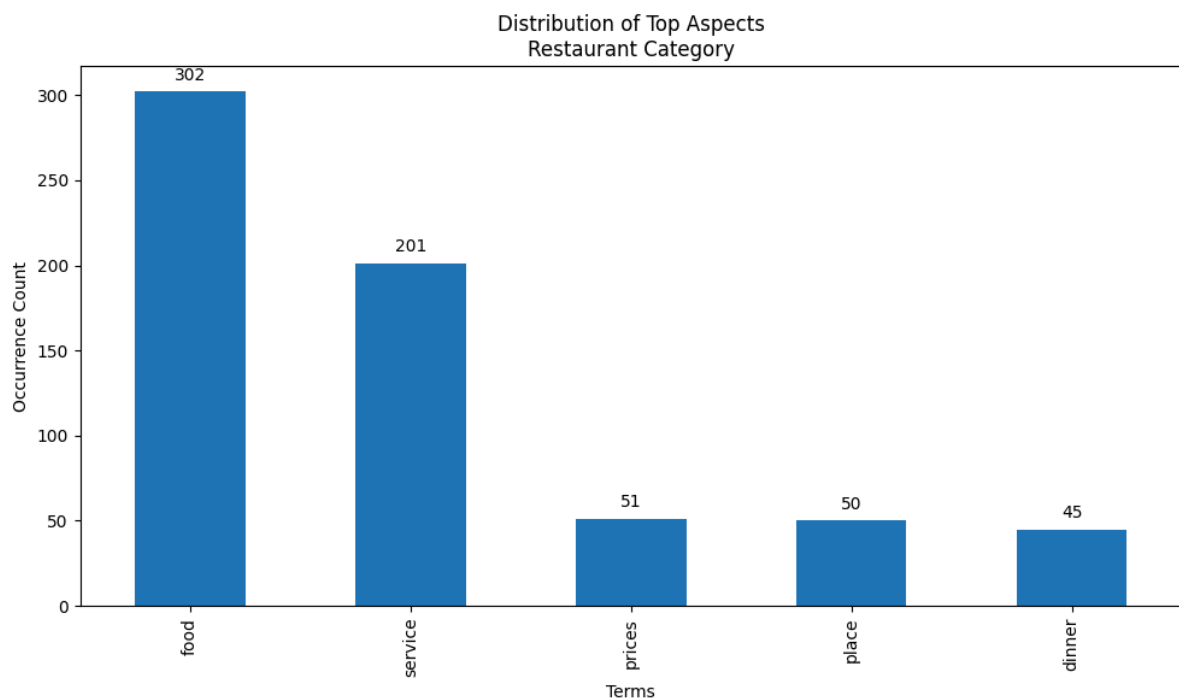
Roll No : 20CS30003

1.Data Analysis

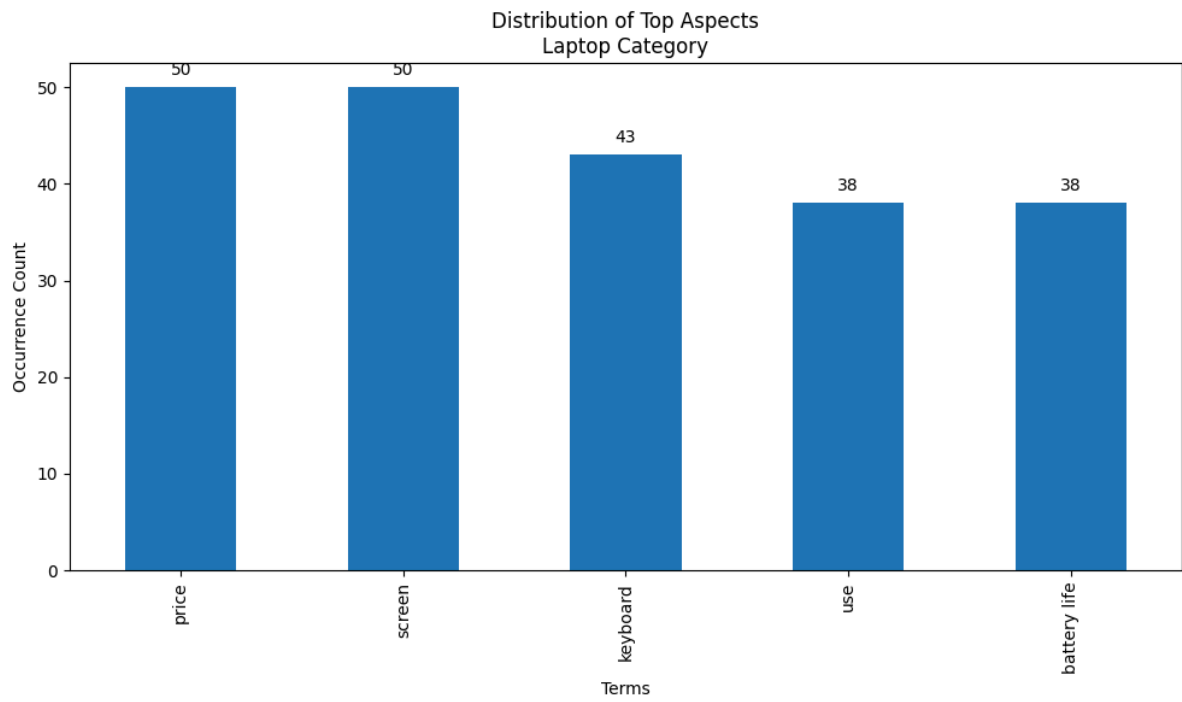
Examines feedback patterns in Restaurant and Laptop categories by:

1. Finding and plotting top 5 common features in each group
2. Showing how many features appear per customer comment

1.1 Key Features Analysis

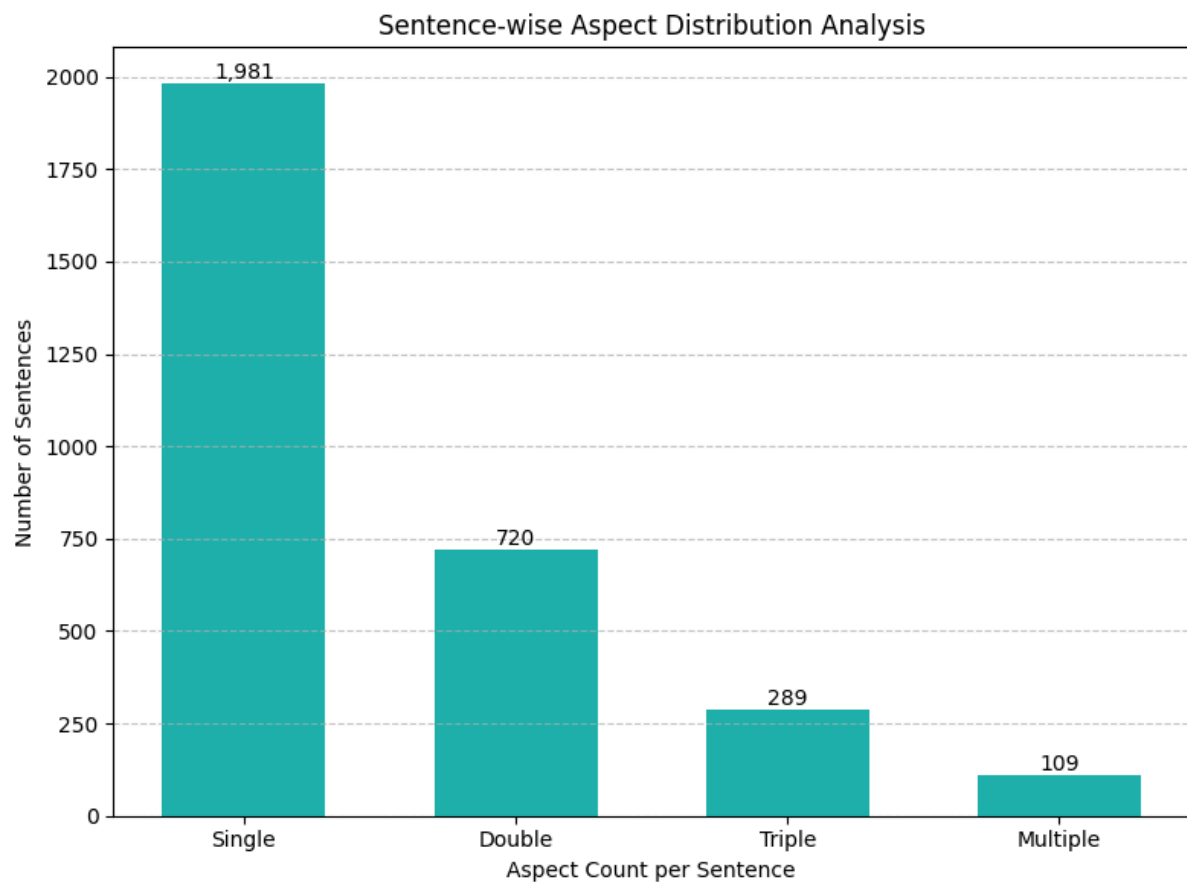


Top 5 most frequently occurring aspects in Restaurant category



Top 5 most frequently occurring aspects in Laptop category

1.2 Distribution of number of aspects per sentence



Distribution of number of aspects per sentence

2. Sentiment Analysis Models

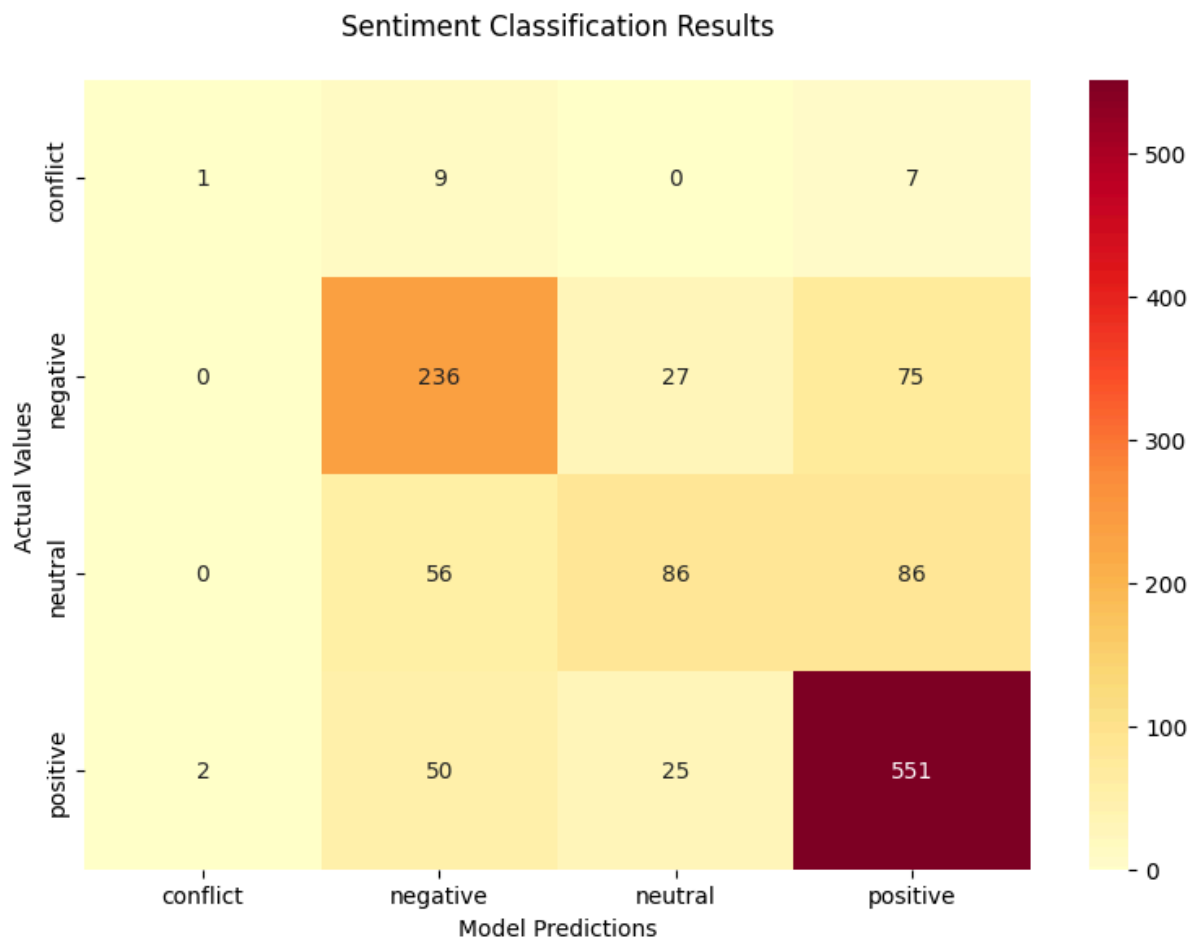
Built two classifiers to analyze feature-level sentiment:

1. SVM classifier
2. BERT (base-uncased)

Input format: sentence + [SEP] + feature term

2.1 Support Vector Machine (SVM) based classifier

Text converted using TF-IDF vectors. Model optimized through parameter testing - optimal values.



Confusion Matrix for SVM-based sentiment classifier.

Parameter grid for SVM hyperparameter tuning.

```
param_grid = {  
    'kernel': ['rbf', 'linear', 'poly'],  
    'C': [0.001, 0.01, 0.1, 1.0, 10.0],  
    'gamma': ['auto', 'scale', 1e-4, 1e-3, 1e-2, 1e-1, 1.0]  
}
```

Optimal parameters found: {

```
    'C': 1.0,  
    'gamma': 'auto',  
    'kernel': 'linear'
```

}

Best performing model configuration:

```
SVC(gamma='auto', kernel='linear')
```

Evaluation Metrics for SVM-based sentiment classifiers.

Model Performance Summary:

```
-----  
Precision Rate:    0.707082  
Recall Rate:      0.721718  
F1 Score:         0.705343  
Accuracy:         0.721718  
-----
```

Table 4: Classification report comparison for SVM model

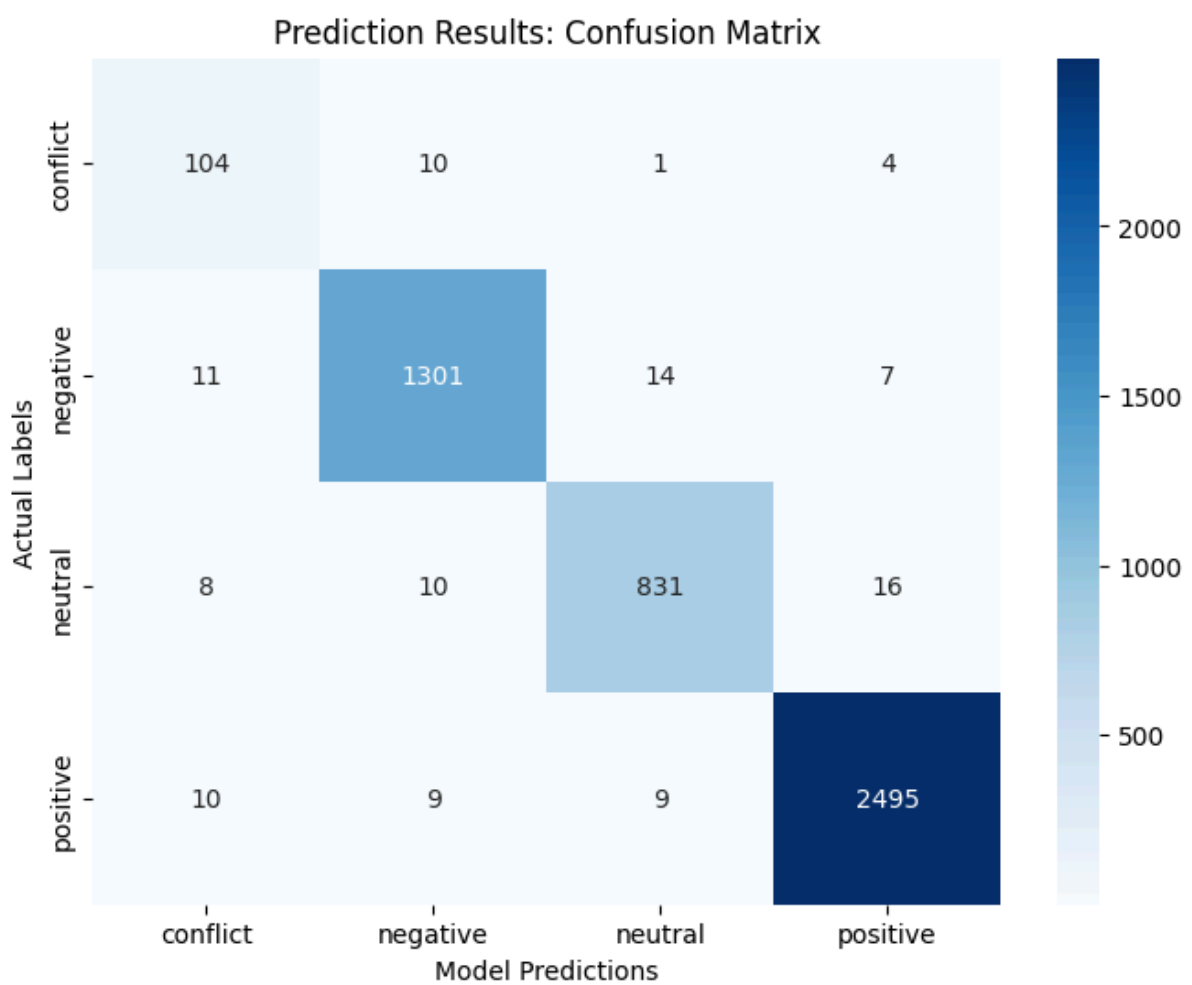
Support Vector Machine Performance Analysis:

	precision	recall	f1-score	support
conflict	0.33	0.06	0.10	17
negative	0.67	0.70	0.69	338
neutral	0.62	0.38	0.47	228
positive	0.77	0.88	0.82	628
accuracy			0.72	1211
macro avg	0.60	0.50	0.52	1211
weighted avg	0.71	0.72	0.71	1211

2.2 BERT-based classifier

BERT Implementation

Used bert-base-uncased with input formatted as: feature + [SEP] + sentence. Model tuned for optimal results.



Confusion Matrix for BERT-based sentiment classifier.

Parameter grid for BERT hyperparameter tuning.

```
def get_hyperparam_space(trial: optuna.Trial) -> dict:
    return {
        'num_train_epochs': trial.suggest_categorical(
            name='num_train_epochs',
            choices=[2, 3, 4]
        ),
        'learning_rate': trial.suggest_float(
            name='learning_rate',
            low=1e-5,
            high=5e-5
        ),
        'per_device_train_batch_size': trial.suggest_categorical(
            name='per_device_train_batch_size',
            choices=[8, 16, 32]
        )
    }
```

Best hyperparameters found:

```
{
    'num_train_epochs': 4,
    'learning_rate': 4.9195827336361806e-05,
    'per_device_train_batch_size': 8
}
```

Evaluation Metrics for BERT-based sentiment classifiers

Model Evaluation Results:

LOSS: 0.0788

ACCURACY: 0.9775

PRECISION: 0.9780

RECALL: 0.9775

F1: 0.9777

RUNTIME: 24.3849

SAMPLES_PER_SECOND: 198.4830

STEPS_PER_SECOND: 12.4260

Classification report comparison for BERT models.

Detailed Performance Analysis:

	precision	recall	f1-score	support
conflict	0.78	0.87	0.83	119
negative	0.98	0.98	0.98	1333
neutral	0.97	0.96	0.97	865
positive	0.99	0.99	0.99	2523
accuracy			0.98	4840
macro avg	0.93	0.95	0.94	4840
weighted avg	0.98	0.98	0.98	4840