

```
In [1]: import random
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from tqdm import tqdm

sns.set()
plt.rcParams['figure.dpi'] = 100
```

Collection of movie reviews

```
In [2]: training_reviews_with_labels = [
    ("Watched the whole thing: surprisingly good!", 'positive'),
    ("Beautifully, thoughtfully made", 'positive'),
    ("A good end to a good season", 'positive'),
    ("Better than expected", 'positive'),
    ("Amazing achievement!!", 'positive'),
    ("It's fantastic", 'positive'),
    ("Fully prepared to hate this... completely surprised!", 'positive'),
    ("The right direction", 'positive'),
    ("A visual and storytelling masterpiece.", 'positive'),
    ("Definitely should see if you're a fan of the genre", 'positive'),
    ("Amazing cinematography, most of the storylines are good", 'positive'),
    ("A great series you can actually watch with your family.", 'positive'),
    ("Pretty good start", 'positive'),

    ("Beautiful to watch, a few interesting characters, storyline is good until it isn't",
    ("So much potential, but less realization", 'neutral'),
    ("Missed the mark", 'neutral'),
    ("Has potential, but also flaws", 'neutral'),
    ("Amazing looks but lacking a clear Jacksonque vision", 'neutral'),
    ("Slow; Slow; Slow", 'neutral'),
    ("It's fine - as in OK - as in mediocre", 'neutral'),
    ("Beautiful but", 'neutral'),
    ("Great CGI but lacks an interesting plot", 'neutral'),

    ("More boring than logic homework", 'negative'),
    ("A major disappointment", 'negative'),
    ("Horrible writing, slow plot, and disrespectful", 'negative'),
    ("An ok fantasy story that has little to do with Tolkien", 'negative'),
    ("Boring, even for generic fantasy", 'negative'),
    ("So many problems", 'negative'),
    ("What is this, an Anti-FanFic or something? Please read the books!", 'negative'),
    ("Just bad all around", 'negative'),
    ("I mourn for what this could have been", 'negative'),
    ("Short version, skip it. You'll know you did right when you don't hear people talking
    ("Could have been so much more", 'negative'),
]
```

```
In [3]: test_reviews_with_labels = [
    ("The best show I've seen so far this year!", 'positive'),
    ("Really enjoyed it", 'positive'),
    ("Amazing.", 'positive'),
    ("Why all the hate? I enjoyed it.", 'positive'),
    ("Beautiful visuals, entertaining, and I believe this show has a lot of potential!",
    ("A beautiful rendering of Middle Earth's history", 'positive'),
    ("So far, so good... and there's still hope", 'positive'),
    ("I'm a fan", 'positive'),
```

```

("It works for me", 'positive'),
("Not the best, but enjoyed every episode. Can't wait to see much much more.", 'positive'),
("Beautiful, flawed, and a wonderful Fall treat", 'positive'),

("Good show with too many subplots", 'neutral'),
("Starts badly, gets better", 'neutral'),
("Good and bad things", 'neutral'),
("Big and beautiful but can use a little help with its energy.", 'neutral'),
("Pretty but ultimately hollow and lacking in engagement", 'neutral'),
("Beautiful visuals and story overshadowed by unnecessary gore and violence", 'neutral'),

("Not what you're probably expecting", 'negative'),
("Poor writing; Uninteresting characters, nonsensical actions.", 'negative'),
("Budget was spent on snacks between shots", 'negative'),
("If you ignore the source material, it's still boring and weird", 'negative'),
("Just a bad show", 'negative'),
("It's awful", 'negative'),
("I was hopeful...", 'negative'),
("Painfully mediocre with a few good spots", 'negative'),
("Beautiful to look at... but that's about it.", 'negative'),
("Underwhelming and disappointing", 'negative'),
("Tolkien is rolling in his grave. No mystery. No inspiration. Wardrobe & acting is p
]

```

In [4]:

```

labels = ['positive', 'neutral', 'negative']

training_reviews = [review for review, label in training_reviews_with_labels]
training_labels = np.array([label for review, label in training_reviews_with_labels])

test_reviews = [review for review, label in test_reviews_with_labels]
test_labels = np.array([label for review, label in test_reviews_with_labels])

print('Labels:')
print(labels)
print()

print(f'There are {len(training_reviews)} training reviews')
print('Training labels:')
print(training_labels)
print()

print(f'There are {len(test_reviews)} test reviews')
print('Test labels:')
print(test_labels)

```

Labels:

```
['positive', 'neutral', 'negative']
```

There are 33 training reviews

Training labels:

```
['positive' 'positive' 'positive' 'positive' 'positive' 'positive'
 'positive' 'positive' 'positive' 'positive' 'positive' 'positive'
 'positive' 'neutral' 'neutral' 'neutral' 'neutral' 'neutral' 'neutral'
 'neutral' 'neutral' 'neutral' 'negative' 'negative' 'negative' 'negative'
 'negative' 'negative' 'negative' 'negative' 'negative' 'negative'
 'negative']
```

There are 28 test reviews

Test labels:

```
['positive' 'positive' 'positive' 'positive' 'positive' 'positive'
 'positive' 'positive' 'positive' 'positive' 'positive' 'neutral'
 'neutral' 'neutral' 'neutral' 'neutral' 'neutral' 'negative' 'negative'
 'negative' 'negative' 'negative' 'negative' 'negative' 'negative'
 'negative' 'negative' 'negative']
```



```

row = {
    'Review': review,
    'Label': label,
    'Prediction': prediction,
}
for label, p in zip(labels, prob):
    row[f'{label} %'] = f'{p * 100:.1f}'

predictions.append(prediction)
table.append(row)

print(f'Accuracy: {accuracy_score(test_labels, predictions) * 100:.1f}%')
print(f'F1 score: {f1_score(test_labels, predictions, labels=labels, average=None)}')

```

Accuracy: 39.3%
F1 score: [0.48 0.30769231 0.33333333]

In [11]:

```

pd.set_option('display.max_rows', None)
pd.set_option('display.max_colwidth', None)
display(pd.DataFrame(table))

```

	Review	Label	Prediction	positive %	neutral %	negative %
0	The best show I've seen so far this year!	positive	positive	63.6	4.9	31.5
1	Really enjoyed it	positive	negative	30.0	33.8	36.2
2	Amazing.	positive	positive	56.6	28.3	15.1
3	Why all the hate? I enjoyed it.	positive	positive	58.8	16.9	24.3
4	Beautiful visuals, entertaining, and I believe this show has a lot of potential!	positive	positive	62.5	22.8	14.7
5	A beautiful rendering of Middle Earth's history	positive	positive	48.4	39.4	12.3
6	So far, so good... and there's still hope	positive	negative	37.5	23.9	38.6
7	I'm a fan	positive	positive	56.3	21.1	22.6
8	It works for me	positive	negative	17.7	21.6	60.7
9	Not the best, but enjoyed every episode. Can't wait to see much much more.	positive	neutral	28.1	59.4	12.5
10	Beautiful, flawed, and a wonderful Fall treat	positive	neutral	33.5	40.9	25.6
11	Good show with too many subplots	neutral	positive	72.0	10.6	17.4
12	Starts badly, gets better	neutral	positive	56.3	21.1	22.6
13	Good and bad things	neutral	positive	72.0	10.6	17.4
14	Big and beautiful but can use a little help with its energy.	neutral	neutral	24.1	61.0	14.9
15	Pretty but ultimately hollow and lacking in engagement	neutral	neutral	9.4	87.6	3.1
16	Beautiful visuals and story overshadowed by unnecessary gore and violence	neutral	negative	32.3	23.2	44.5
17	Not what you're probably expecting	negative	negative	38.5	5.7	55.8
18	Poor writing; Uninteresting characters, nonsensical actions.	negative	neutral	24.1	39.2	36.7
19	Budget was spent on snacks between shots	negative	positive	39.4	27.3	33.3
20	If you ignore the source material, it's still boring and weird	negative	positive	55.6	3.1	41.3

	Review	Label	Prediction	positive %	neutral %	negative %
21	Just a bad show	negative	negative	20.6	16.7	62.7
22	It's awful	negative	negative	30.0	33.8	36.2
23	I was hopeful...	negative	positive	39.4	27.3	33.3
24	Painfully mediocre with a few good spots	negative	positive	57.1	36.4	6.5
25	Beautiful to look at... but that's about it.	negative	neutral	4.2	85.3	10.4
26	Underwhelming and disappointing	negative	positive	45.9	17.2	36.9
27	Tolkien is rolling in his grave. No mystery. No inspiration. Wardrobe & acting is pretty bland.	negative	neutral	10.2	63.3	26.5

In [12]:

```
mat = confusion_matrix(test_labels, predictions, labels=labels)

sns.heatmap(mat, square=True, annot=True, xticklabels=labels, yticklabels=labels,
            linewidths=1)
plt.xlabel("Model predictions")
plt.ylabel("True labels")
plt.show()
```

