

```
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
import torch
import transformers

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 Jacksonesque 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.", 'positi
("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 pi
]

```

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'
```

# Language models

```
In [5]: from transformers import AutoModelForCausalLM, AutoTokenizer

transformers.utils.logging.set_verbosity_error()
```

```
In [6]: models = {}
tokenizers = {}
```

```
In [7]: models['GPT-2'] = AutoModelForCausalLM.from_pretrained('gpt2-large').to('cuda:0')
tokenizers['GPT-2'] = AutoTokenizer.from_pretrained('gpt2-large')

print('Number of parameters (GPT-2 large):', sum(p.numel() for p in models['GPT-2'].parameters()))

Number of parameters (GPT-2 large): 774030080
```

```
In [8]: models['GPT-Neo'] = AutoModelForCausalLM.from_pretrained('EleutherAI/gpt-neo-2.7B').to('cuda:0')
tokenizers['GPT-Neo'] = AutoTokenizer.from_pretrained('EleutherAI/gpt-neo-2.7B')

print('Number of parameters (GPT-Neo 2.7B):', sum(p.numel() for p in models['GPT-Neo'].parameters()))

Number of parameters (GPT-Neo 2.7B): 2651307520
```

```
In [9]: models['GPT-J'] = AutoModelForCausalLM.from_pretrained(
    "EleutherAI/gpt-j-6B", revision="float16", torch_dtype=torch.float16, low_cpu_mem_usage=True
).to('cuda:2')
tokenizers['GPT-J'] = AutoTokenizer.from_pretrained("EleutherAI/gpt-j-6B")

print('Number of parameters (GPT-J 6B):', sum(p.numel() for p in models['GPT-J'].parameters()))

Number of parameters (GPT-J 6B): 6050882784
```

```
In [10]: def generate(prompt, model_name='GPT-2', exclude_prompt=False, **kwargs):
    model = models[model_name]
    tokenizer = tokenizers[model_name]

    x = tokenizer(prompt, return_tensors='pt').input_ids.to(model.device)
    y = model.generate(x, **kwargs).cpu()
    return tokenizer.batch_decode(y)[0][len(prompt) if exclude_prompt else 0:]
```

```
In [11]: transformers.set_seed(1)
print(generate('Once upon a time in Bologna', max_new_tokens=38, do_sample=True, top_p=0.95))

Once upon a time in Bologna, Italy, a man named Cesare Borgia had a problem.

He was the king of the most populous nation in the world, and his country was, quite frankly, a
```

```
In [12]: transformers.set_seed(2)
print(generate('Once upon a time in Bologna', model_name='GPT-Neo', max_new_tokens=100, do_sample=True, top_p=0.95))

Once upon a time in Bologna, the capital of Italy, a boy played football on the streets in the evenings with an old leather ball. The ball had been given him by an admirer of his, a
```

nd he never tired of using it. He liked to kick it about, kick it over the houses, kick it through the windows, kick it through people. Sometimes he would run with it over the walls of houses and into the water of the streets. In a way he was trying to win the admiration of his admirer

In [13]:

```
transformers.set_seed(3)
print(generate('Once upon a time in Bologna', model_name='GPT-J', max_new_tokens=100, do_s
```

Once upon a time in Bologna

We met for the first time in the bar. It was a Wednesday in the spring of 2017, and we were both students at the University of Bologna, although we didn't know that yet. We just started talking and I guess we were both trying to convince the other that he wasn't completely mad. We found out pretty soon that we were both interested in the same topic: the history of Italian cinema. We spent the rest of the evening exchanging e-

In [14]:

```
prompt = '''Midway upon the journey of our life
I found myself within a forest dark,
For the straightforward pathway had been lost'''

for model_name in models:
    transformers.set_seed(4)
    print(generate(prompt, model_name=model_name, max_new_tokens=100, do_sample=True,
                  top_p=0.9))
    print('-----')
```

```
Midway upon the journey of our life
I found myself within a forest dark,
For the straightforward pathway had been lost:
I saw, as I looked upon the trees,
I saw the faces of all the men who are
Of our time and place,
And the men they were, and their lives, and their hearts.
Then, with a feeling that I had never felt before,
I knew it was the road
I was on.
The trees, the hills, the valleys, the trees, the hills, the valleys, the trees, the hills
are the roads to our
-----
Midway upon the journey of our life
I found myself within a forest dark,
For the straightforward pathway had been lost
and so I had turned back to where
I knew the road would lead.
I saw two travelers,
Each alone,
And so I spoke with them
And I asked them
For my life in return
For what I asked I found that they would share with me

A path you don't know will lead you to your
destiny and your purpose,
You'll learn how to speak to those who won't listen to you,
and you'll know how
-----
Midway upon the journey of our life
I found myself within a forest dark,
For the straightforward pathway had been lost.

''Turn back,' said the lady, 'turn back with me,
Turn back, lest you lose your way with me.'
So I turned back to the straight path that I found
```

''I am sorry,' said the lady, 'sorry for you,  
And sorry that you are in trouble now.'  
For I lost the way that was good to me

''

-----

In [15]:

```
prompt = '''All that is gold does not glitter,  
Not all those who wander are lost;  
The old that is strong does not wither,  
Deep roots are not reached by the frost.  
A dawn'''\n\nfor model_name in models:  
    transformers.set_seed(5)  
    print(generate(prompt, model_name=model_name, max_new_tokens=100, do_sample=True,  
                  top_p=0.9))  
    print('-----')
```

All that is gold does not glitter,  
Not all those who wander are lost;  
The old that is strong does not wither,  
Deep roots are not reached by the frost.  
A dawn from above comes to me,  
And bright and glorious it shines;  
The sun's light does not set,  
Not the moon does not wane.  
The days are old, the years are short;  
The mountains lie still, the rocks are high;  
The hills are no more, and the clouds,  
The clouds no more.  
A song comes to me from beyond the grave,  
I know, because the voice sings of joy;  
It is I that am old

-----

All that is gold does not glitter,  
Not all those who wander are lost;  
The old that is strong does not wither,  
Deep roots are not reached by the frost.  
A dawn breaking in the sky  
Is the chance of a lifetime;  
The stars that are hidden do not die,  
The life that is whole is not destroyed.

Saturday, April 10, 2010

One morning, early one morning while I was sleeping, I woke up with the sun shining directly into my face, a nice refreshing shower, I could see the blue sky through the window and it was a pretty sight, just the two of us and our bed. I was getting up,

-----

All that is gold does not glitter,  
Not all those who wander are lost;  
The old that is strong does not wither,  
Deep roots are not reached by the frost.  
A dawn sun, a spring rain, a summer's sun, they bear thee away,  
Nor do they care if, like the grass,  
Thou wilt grow up again.  
The place where thou wast laid to sleep is covered,  
Not alone with marble;  
Thy Mother Earth received thee, and her face is ever turning  
Toward the hills, and toward the sky.

I have been in search of the ideal, of the great, of the magnificent.  
And

-----

In [16]:

```
prompt = '''Theorem: All groups of order 4 are abelian.
Proof:'''

for model_name in models:
    transformers.set_seed(5)
    print(generate(prompt, model_name=model_name, max_new_tokens=120))
    print('-----')
```

Theorem: All groups of order 4 are abelian.

Proof: Let  $G$  be a group of order 4. Then  $G$  is a group of order 4, and  $G$  is a group of order 4.

Theorem: All groups of order 5 are abelian.

Proof: Let  $G$  be a group of order 5. Then  $G$  is a group of order 5, and  $G$  is a group of order 5.

Theorem: All groups of order 6 are abelian.

Proof: Let  $G$  be a group of order 6. Then  $G$  is a group of order

-----  
Theorem: All groups of order 4 are abelian.

Proof:

Let  $G$  be a group of order 4.

Then  $G$  is abelian.

Proof:

Let  $x, y, z, t$  be elements of  $G$ .

Then  $x=y=z=t$ .

Proof:

Let  $x, y, z, t$  be elements of  $G$ .

Then  $x=y=z=t$ .

Proof:

Let  $x, y, z, t$  be elements of  $G$ .

Then  $x=y=z=t$ .

Proof:

Let

-----  
Theorem: All groups of order 4 are abelian.

Proof: Let  $G$  be a group of order 4. Then  $G$  is isomorphic to  $\mathbb{Z}_4$  or  $\mathbb{Z}_2 \times \mathbb{Z}_2$ .

If  $G \cong \mathbb{Z}_4$ , then  $G$  is abelian.

If  $G \cong \mathbb{Z}_2 \times \mathbb{Z}_2$ , then  $G$  is abelian.

A:

Theorem: If  $G$  is

-----

In [17]:

```
prompt = '''Theorem: All groups of order 6 are abelian.
Proof:'''

for model_name in models:
    transformers.set_seed(5)
    print(generate(prompt, model_name=model_name, max_new_tokens=120))
    print('-----')
```

Theorem: All groups of order 6 are abelian.

Proof: Let  $G$  be a group of order 6. Then  $G$  is a group of order 6, and  $G$  is a group of order 6.

Theorem: All groups of order 7 are abelian.

Proof: Let  $G$  be a group of order 7. Then  $G$  is a group of order 7, and  $G$  is a group of order 7.

Theorem: All groups of order 8 are abelian.

Proof: Let  $G$  be a group of order 8. Then  $G$  is a group of order

-----  
Theorem: All groups of order 6 are abelian.

Proof:

Let  $GG$  be a group of order 6.

Then  $GG$  is abelian.

Proof:

Let  $x, y, z, t$  be elements of  $GG$ .

Then  $x=y=z=t$ .

Proof:

Let  $x, y, z, t$  be elements of  $GG$ .

Then  $x=y=z=t$ .

Proof:

Let  $x, y, z, t$  be elements of  $GG$ .

Then  $x=y=z=t$ .

Proof:

Let

-----  
Theorem: All groups of order 6 are abelian.

Proof: Let  $GG$  be a group of order 6. Then  $GG$  is cyclic of order 6. Let  $GG$  be generated by  $g$ . Then  $g^6=1$ . Hence  $g^3=1$ . Hence  $g^2=1$ . Hence  $g=1$ . Hence  $GG$  is abelian.

A:

Theorem: Let  $GG$  be a group of order  $6$ . Then  $GG$  is abelian.

Proof: Let  $GG$  be a group of order  $6$ . Then  $GG$

## In-context learning

In [18]:

```
def construct_prompt(review):
    return f'Review: "{review}"\nSentiment:'

def construct_prompt_with_examples(test_review):
    return '\n'.join(
        construct_prompt(training_review) + " " + label
        for training_review, label in sorted(training_reviews_with_labels, key=lambda _: 1)
    ) + '\n' + construct_prompt(test_review)
```

In [19]:

```
test_review = test_reviews[-1]
prompt = construct_prompt_with_examples(test_review)
res = generate(prompt, max_new_tokens=1)
print(res)
```

Review: "Boring, even for generic fantasy"

Sentiment: negative

Review: "Has potential, but also flaws"

Sentiment: neutral

Review: "Fully prepared to hate this... completely surprised!"

Sentiment: positive

Review: "A great series you can actually watch with your family."

Sentiment: positive

Review: "An ok fantasy story that has little to do with Tolkien"

Sentiment: negative

Review: "A major disappointment"

Sentiment: negative

Review: "Great CGI but lacks an interesting plot"

Sentiment: neutral

Review: "What is this, an Anti-FanFic or something? Please read the books!"

Sentiment: negative

Review: "Just bad all around"

Sentiment: negative

Review: "Amazing looks but lacking a clear Jacksoneque vision"

Sentiment: neutral

Review: "Beautiful to watch, a few interesting characters, storyline is good until it is n't"

Sentiment: neutral  
Review: "Slow; Slow; Slow"  
Sentiment: neutral  
Review: "Could have been so much more"  
Sentiment: negative  
Review: "The right direction"  
Sentiment: positive  
Review: "Pretty good start"  
Sentiment: positive  
Review: "So much potential, but less realization"  
Sentiment: neutral  
Review: "Missed the mark"  
Sentiment: neutral  
Review: "Horrible writing, slow plot, and disrespectful"  
Sentiment: negative  
Review: "Watched the whole thing: surprisingly good!"  
Sentiment: positive  
Review: "Definitely should see if you're a fan of the genre"  
Sentiment: positive  
Review: "Amazing achievement!!"  
Sentiment: positive  
Review: "Beautifully, thoughtfully made"  
Sentiment: positive  
Review: "Beautiful but"  
Sentiment: neutral  
Review: "A good end to a good season"  
Sentiment: positive  
Review: "More boring than logic homework"  
Sentiment: negative  
Review: "So many problems"  
Sentiment: negative  
Review: "Short version, skip it. You'll know you did right when you don't hear people talking about it."  
Sentiment: negative  
Review: "Amazing cinematography, most of the storylines are good"  
Sentiment: positive  
Review: "It's fine - as in OK - as in mediocre"  
Sentiment: neutral  
Review: "It's fantastic"  
Sentiment: positive  
Review: "Better than expected"  
Sentiment: positive  
Review: "A visual and storytelling masterpiece."  
Sentiment: positive  
Review: "I mourn for what this could have been"  
Sentiment: negative  
Review: "Tolkien is rolling in his grave. No mystery. No inspiration. Wardrobe & acting is pretty bland."  
Sentiment: negative

In [20]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
from IPython.display import display

transformers.set_seed(0)

predictions = {model_name: [] for model_name in models}
table = []

for review, label in tqdm(zip(test_reviews, test_labels), total=len(test_reviews)):
    prompt = construct_prompt_with_examples(review)

    row = {
        'Review': review,
        'Label': label,
    }
```



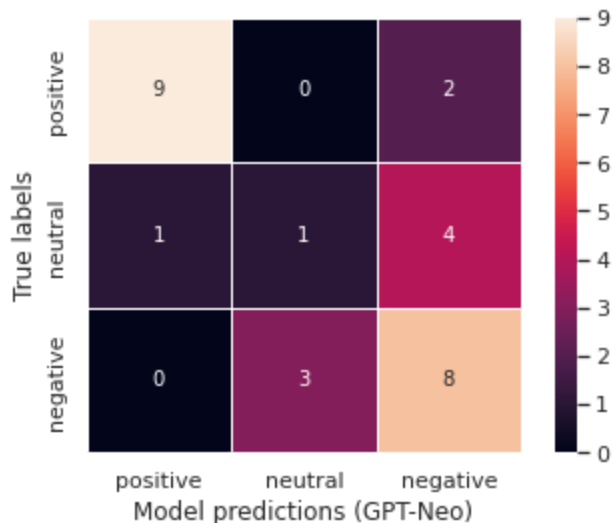
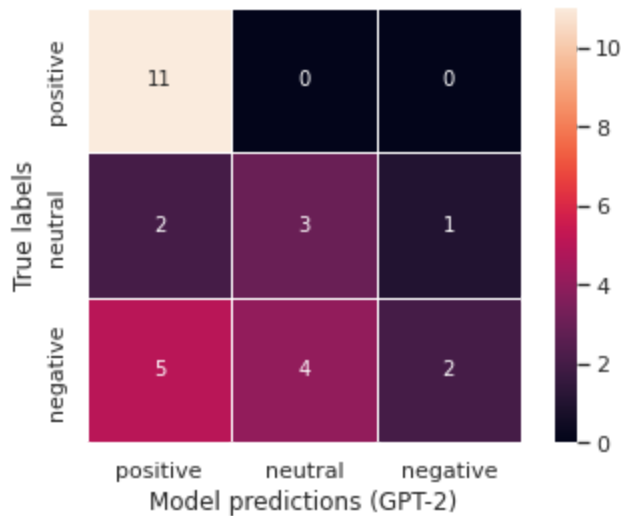


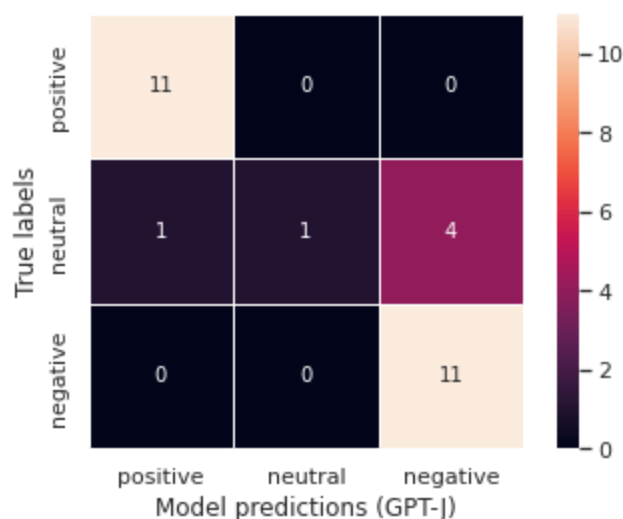
	Review	Label	Prediction (GPT-2)	Prediction (GPT-Neo)	Prediction (GPT-J)
18	Poor writing; Uninteresting characters, nonsensical actions.	negative	positive	negative	negative
19	Budget was spent on snacks between shots	negative	neutral	negative	negative
20	If you ignore the source material, it's still boring and weird	negative	positive	negative	negative
21	Just a bad show	negative	neutral	negative	negative
22	It's awful	negative	negative	negative	negative
23	I was hopeful...	negative	positive	neutral	negative
24	Painfully mediocre with a few good spots	negative	neutral	neutral	negative
25	Beautiful to look at... but that's about it.	negative	positive	neutral	negative
26	Underwhelming and disappointing	negative	negative	negative	negative
27	Tolkien is rolling in his grave. No mystery. No inspiration. Wardrobe & acting is pretty bland.	negative	positive	negative	negative

In [22]:

```
for model_name, model_predictions in predictions.items():
    mat = confusion_matrix(test_labels, model_predictions, labels=labels)

    sns.heatmap(mat, square=True, annot=True, xticklabels=labels, yticklabels=labels, line
plt.xlabel(f"Model predictions ({model_name})")
plt.ylabel("True labels")
plt.show()
```





## Chain of thought

In [23]:

```
prompt = '''Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls does he have now?
A: The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many
```

In [24]:

```
for model_name in models:
    print(generate(prompt, model_name=model_name, max_new_tokens=10))
    print('-----')
```

```
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
A: The answer is 6.
```

```
Q
-----
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
A: The answer is 6.
```

```
Q
-----
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
A: The answer is 17.
```

In [25]:

```
prompt = '''Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many
```

In [26]:

```
for model_name in models:
    print(generate(prompt, model_name=model_name, max_new_tokens=20))
    print('-----')
```

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many

tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A:  $23 - 20 + 6 = 9$ .

Q: Roger has a tennis ball.

-----

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many

tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 23.

Q: The cafeteria had 23 apples. If they used

-----

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many

tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples.  $20 + 6 = 26$ . The answer is 26.

-----

In [27]:

```
def construct_prompt_cot(review):
    return f'''Q: What is the sentiment of this movie review? "Beautifully, thoughtfully r
A: The reviewer is very happy about how the movie was made. The sentiment is positive.
Q: What is the sentiment of this movie review? "Boring, even for generic fantasy"
A: The reviewer is bored by the movie. The sentiment is negative.
Q: What is the sentiment of this movie review? "It's fine - as in OK - as in mediocre"
A: The reviewer does not feel particularly good or bad about the movie. The sentiment is r
Q: What is the sentiment of this movie review? "{review}"'''
```

In [28]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
from IPython.display import display

predictions = {model_name: [] for model_name in models}
full_predictions = {model_name: [] for model_name in models}
table = []

for review, label in tqdm(zip(test_reviews, test_labels), total=len(test_reviews)):
    prompt = construct_prompt_cot(review)

    row = {
        'Review': review,
        'Label': label,
    }

    for model_name in models:
        res = generate(prompt, model_name=model_name, max_new_tokens=30)
        full_prediction = res.split('\n')[7].split('A: ')[1]
        prediction = full_prediction.split()[-1].strip('.')
```



	Review	Label	Prediction (GPT-2)	Full prediction (GPT-2)	Prediction (GPT-Neo)	Full prediction (GPT-Neo)	Prediction (GPT-J)	Full prediction (GPT-J)
4	Beautiful visuals, entertaining, and I believe this show has a lot of potential!	positive	positive	The reviewer is very happy about the movie. The sentiment is positive.	positive	The reviewer is very happy about the visuals, the entertainment, and the potential of the show. The sentiment is positive.	positive	The reviewer is very happy about the movie. The sentiment is positive.
5	A beautiful rendering of Middle Earth's history	positive	positive	The reviewer is happy about the movie. The sentiment is positive.	positive	The reviewer is very happy about how the movie was made. The sentiment is positive.	positive	The reviewer is very happy about how the movie was made. The sentiment is positive.
6	So far, so good... and there's still hope	positive	positive	The reviewer is very happy about the movie. The sentiment is positive.	positive	The reviewer is very hopeful about the future of the movie. The sentiment is positive.	positive	The reviewer is optimistic about the movie. The sentiment is positive.
7	I'm a fan	positive	positive	The reviewer is a fan of the movie. The sentiment is positive.	positive	The reviewer is a fan of the movie. The sentiment is positive.	positive	The reviewer is a fan of the movie. The sentiment is positive.
8	It works for me	positive	positive	The reviewer is happy with the movie. The sentiment is positive.	positive	The reviewer likes the movie. The sentiment is positive.	positive	The reviewer is happy with the movie. The sentiment is positive.
9	Not the best, but enjoyed every episode. Can't wait to see much more.	positive	positive	The reviewer is very happy about the movie. The sentiment is positive.	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	positive	The reviewer is happy about the movie. The sentiment is positive.
10	Beautiful, flawed, and a wonderful Fall treat	positive	positive	The reviewer is very happy about the movie. The sentiment is positive.	positive	The reviewer is very happy about the movie. The sentiment is positive.	positive	The reviewer is very happy about the movie. The sentiment is positive.
11	Good show with too many subplots	neutral	negative	The reviewer is not happy with the movie. The sentiment is negative.	positive	The reviewer is happy with the movie. The sentiment is positive.	positive	The reviewer is happy with the movie. The sentiment is positive.

	Review	Label	Prediction (GPT-2)	Full prediction (GPT-2)	Prediction (GPT-Neo)	Full prediction (GPT-Neo)	Prediction (GPT-J)	Full prediction (GPT-J)
12	Starts badly, gets better	neutral	negative	The reviewer is not happy with the movie. The sentiment is negative.	positive	The reviewer is not happy with the movie. The sentiment is positive.	neutral	The reviewer is not sure about the movie. The sentiment is neutral.
13	Good and bad things	neutral	positive	The reviewer is happy about the movie. The sentiment is positive.	positive	The reviewer is happy about the good things in the movie. The sentiment is positive.	positive	The reviewer is happy about the movie. The sentiment is positive.
14	Big and beautiful but can use a little help with its energy.	neutral	positive	The reviewer is happy about the movie. The sentiment is positive.	positive	The reviewer is very happy about the movie. The sentiment is positive.	positive	The reviewer is happy with the movie. The sentiment is positive.
15	Pretty but ultimately hollow and lacking in engagement	neutral	negative	The reviewer is not particularly happy with the movie. The sentiment is negative.	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	negative	The reviewer is disappointed by the movie. The sentiment is negative.
16	Beautiful visuals and story overshadowed by unnecessary gore and violence	neutral	negative	The reviewer is not happy with the movie. The sentiment is negative.	positive	The reviewer is very happy about the visuals and story. The sentiment is positive.	positive	The reviewer is very happy about the movie. The sentiment is positive.
17	Not what you're probably expecting	negative	positive	The reviewer is not sure what the movie is about. The sentiment is positive.	neutral	The reviewer is not what you are probably expecting. The sentiment is neutral.	negative	The reviewer is not happy with the movie. The sentiment is negative.
18	Poor writing; Uninteresting characters, nonsensical actions.	negative	neutral	The reviewer does not feel particularly good or bad about the movie. The sentiment is neutral.	negative	The reviewer is disappointed by the writing. The sentiment is negative.	negative	The reviewer is disappointed by the movie. The sentiment is negative.
19	Budget was spent on snacks between shots	negative	negative	The reviewer is not happy about the movie. The sentiment is negative.	positive	The reviewer is very happy about the budget. The sentiment is positive.	positive	The reviewer is happy about the movie. The sentiment is positive.

	Review	Label	Prediction (GPT-2)	Full prediction (GPT-2)	Prediction (GPT-Neo)	Full prediction (GPT-Neo)	Prediction (GPT-J)	Full prediction (GPT-J)
20	If you ignore the source material, it's still boring and weird	negative	positive	The reviewer is not sure about the movie. The sentiment is positive.	negative	The reviewer is bored by the source material. The sentiment is negative.	negative	The reviewer is not impressed by the movie. The sentiment is negative.
21	Just a bad show	negative	neutral	The reviewer does not feel particularly good or bad about the movie. The sentiment is neutral.	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	negative	The reviewer is disappointed by the movie. The sentiment is negative.
22	It's awful	negative	negative	The reviewer is very unhappy about the movie. The sentiment is negative.	negative	The reviewer is very disappointed by the movie. The sentiment is negative.	negative	The reviewer is very unhappy about the movie. The sentiment is negative.
23	I was hopeful...	negative	positive	The reviewer is hopeful about the movie. The sentiment is positive.	positive	The reviewer was hopeful about the movie. The sentiment is positive.	positive	The reviewer is hopeful about the movie. The sentiment is positive.
24	Painfully mediocre with a few good spots	negative	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	negative	The reviewer is not happy with the movie. The sentiment is negative.
25	Beautiful to look at... but that's about it.	negative	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	positive	The reviewer is very happy about the movie. The sentiment is positive.	negative	The reviewer is not impressed by the movie. The sentiment is negative.
26	Underwhelming and disappointing	negative	negative	The reviewer is disappointed by the movie. The sentiment is negative.	negative	The reviewer is disappointed by the movie. The sentiment is negative.	negative	The reviewer is disappointed by the movie. The sentiment is negative.
27	Tolkien is rolling in his grave. No mystery. No inspiration. Wardrobe & acting is pretty bland.	negative	negative	The reviewer is not impressed by the movie. The sentiment is negative.	negative	The reviewer is not particularly happy about the movie. The sentiment is negative.	negative	The reviewer is not happy with the movie. The sentiment is negative.



In [30]:

```
for model_name, model_predictions in predictions.items():  
    mat = confusion_matrix(test_labels, model_predictions, labels=labels)  
  
    sns.heatmap(mat, square=True, annot=True, xticklabels=labels, yticklabels=labels, line  
plt.xlabel(f"Model predictions ({model_name})")  
plt.ylabel("True labels")  
plt.show()
```

