Choosing the Right Model:

After training models using different parameters I will compare their training results in order to chose one for the project.

Importing the necessary things:

In [1]:
```
import tabulate as tabulate
from data_repository.sqlite_functions import create_connection
from IPython.display import HTML, display
import matplotlib.pyplot as plt
```

Setting up the query and connection:

To train the models I used different epochs but it seemed that they didn't do much as a result I will only be comparing models with 5 epochs.

In [2]:
```
result_query = """  SELECT * FROM training_results
                    WHERE epoch = 5 """

conn = create_connection("./data_repository/dataset.db")
cur = conn.cursor()
cur.execute(result_query)
rows = cur.fetchall()

rows_lr_01 = list(filter(lambda x: x[2] == '0.1', rows))
rows_lr_05 = list(filter(lambda x: x[2] == '0.5', rows))
rows_lr_10 = list(filter(lambda x: x[2] == '1.0', rows))
```

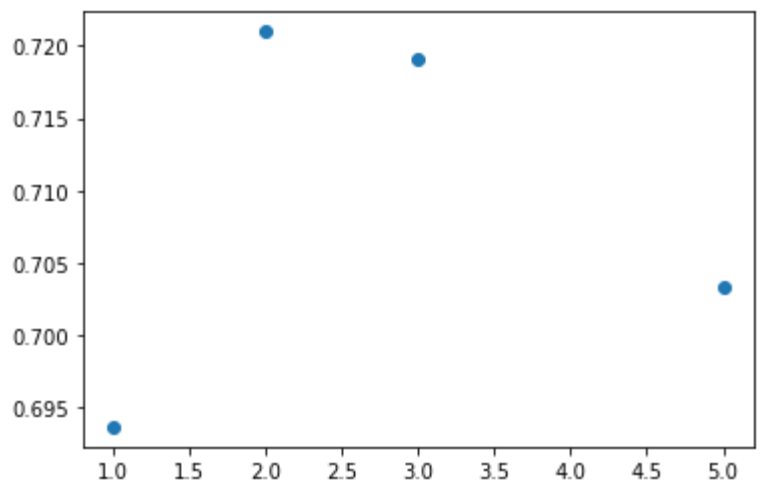
Plotting the different models:

In [3]:
```
display(HTML(tabulate.tabulate(rows, tablefmt='html')))
```

1	5	0.1	1	0.693638	0.693638
2	5	0.1	3	0.719124	0.719124
3	5	0.1	5	0.703297	0.703297
4	5	0.5	1	0.694619	0.694619
5	5	0.5	3	0.708358	0.708358
6	5	0.5	5	0.687286	0.687286
7	5	1	1	0.694242	0.694242
8	5	1	3	0.705912	0.705912
9	5	1	5	0.685555	0.685555
11	5	0.1	2	0.721	0.721

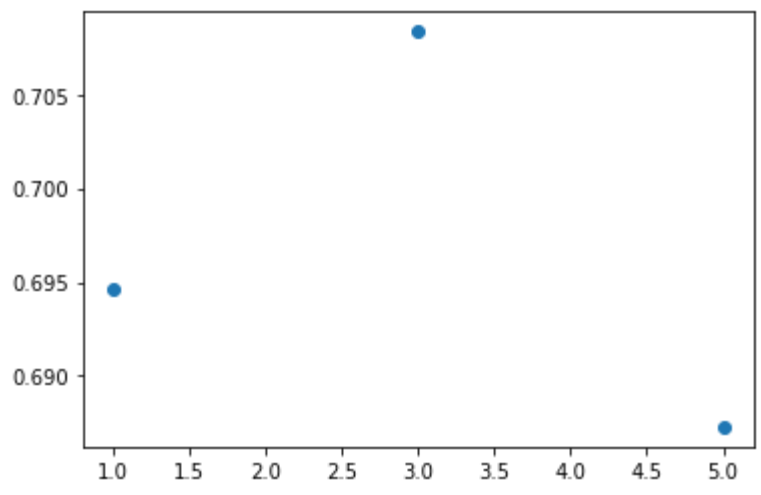
Where learning rate = 0.1 and epoch = 5:

In [4]:
```
x = list(map(lambda item : item[3], rows_lr_01))
y = list(map(lambda item : float(item[4]), rows_lr_01))

fig = plt.figure()
ax = fig.add_subplot()
ax.scatter(x, y)
plt.show()
```



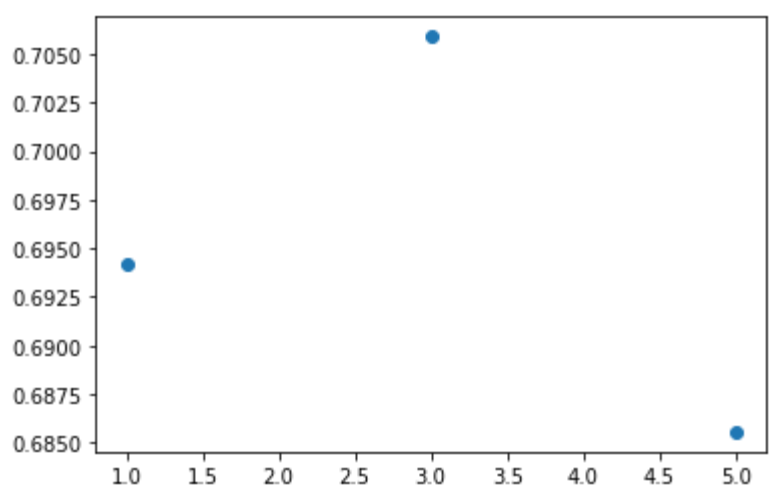
Where learning rate = 0.5 and epoch = 5:

In [5]:
```
x1 = list(map(lambda item : item[3], rows_lr_05))
y1 = list(map(lambda item : float(item[4]), rows_lr_05))

fig = plt.figure()
ax = fig.add_subplot()
ax.scatter(x1, y1)
plt.show()
```



Where learning rate = 1.0 and epoch = 5:

In [6]:
```
x2 = list(map(lambda item : item[3], rows_lr_10))
y2 = list(map(lambda item : float(item[4]), rows_lr_10))

fig = plt.figure()
ax = fig.add_subplot()
ax.scatter(x2, y2)
plt.show()
```



Conclusion:

After comparing the different models it seems that the highest accuracy achieved is 72.1 % using a learning rate of 0.1, 5 epochs and 2 word n grams.