In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [5]:
```
labels = {
    1: 'nenhuma técnica',
    2: 'stemm',
    3: 'lemma',
    5: 'stopword',
    4: 'stemm + lemma',
    6: 'stopword + stemm',
    7: 'stopword + lemma',
    8: 'todas as técnincas'
}
```

In [6]:
```
recomendations_1 = pd.read_csv('../result/alternative/recomendations_1.csv')
recomendations_2 = pd.read_csv('../result/alternative/recomendations_2.csv')
recomendations_3 = pd.read_csv('../result/alternative/recomendations_3.csv')
recomendations_4 = pd.read_csv('../result/alternative/recomendations_4.csv')
recomendations_5 = pd.read_csv('../result/alternative/recomendations_5.csv')
recomendations_6 = pd.read_csv('../result/alternative/recomendations_6.csv')
recomendations_7 = pd.read_csv('../result/alternative/recomendations_7.csv')
recomendations_8 = pd.read_csv('../result/alternative/recomendations_8.csv')
```

In [7]:
```
recomendations_1.head()
```

Out[7]:

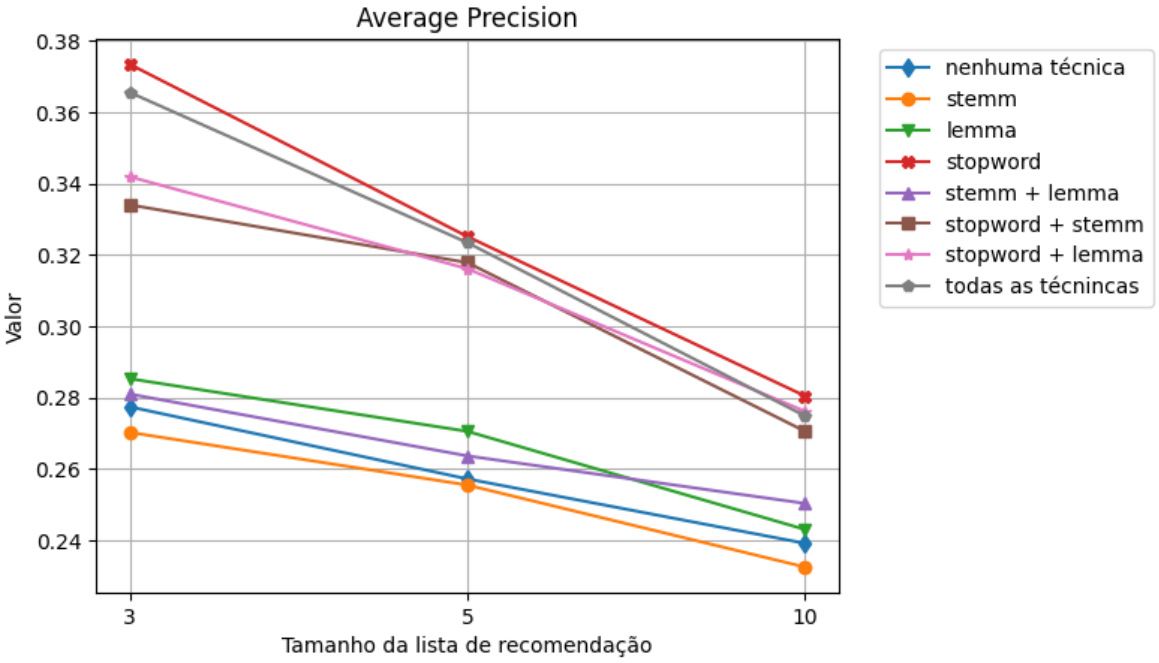
	user_id	prc_10	prc_5	prc_3	ap_10	ap_5	ap_3	rr_10	rr_5	
0	1	0.2	0.2	0.333333	0.326508	0.456667	0.611111	1.000000	1.000000	1
1	4	0.2	0.4	0.666667	0.435794	0.613333	0.722222	1.000000	1.000000	1
2	5	0.3	0.4	0.666667	0.369405	0.413333	0.388889	0.500000	0.500000	0
3	6	0.1	0.2	0.333333	0.142897	0.156667	0.111111	0.333333	0.333333	0
4	7	0.1	0.2	0.333333	0.192897	0.256667	0.277778	0.500000	0.500000	0

In [8]:
```
x = [3, 5, 10]
x2 = np.arange(len(x))
```

In [9]:
```
y_rec_1_prc = [recomendations_1.prc_3.mean(), recomendations_1.prc_5.mean(), rec
y_rec_2_prc = [recomendations_2.prc_3.mean(), recomendations_2.prc_5.mean(), rec
y_rec_3_prc = [recomendations_3.prc_3.mean(), recomendations_3.prc_5.mean(), rec
y_rec_4_prc = [recomendations_4.prc_3.mean(), recomendations_4.prc_5.mean(), rec
y_rec_5_prc = [recomendations_5.prc_3.mean(), recomendations_5.prc_5.mean(), rec
y_rec_6_prc = [recomendations_6.prc_3.mean(), recomendations_6.prc_5.mean(), rec
y_rec_7_prc = [recomendations_7.prc_3.mean(), recomendations_7.prc_5.mean(), rec
y_rec_8_prc = [recomendations_8.prc_3.mean(), recomendations_8.prc_5.mean(), rec
```

In [10]:
```
plt.title("Average Precision")
plt.grid()
plt.plot(x2, y_rec_1_prc, label=labels[1], marker='d')
plt.plot(x2, y_rec_2_prc, label=labels[2], marker='o')
plt.plot(x2, y_rec_3_prc, label=labels[3], marker='v')
plt.plot(x2, y_rec_5_prc, label=labels[5], marker='X')
```

```
plt.plot(x2, y_rec_4_prc, label=labels[4], marker='^')
plt.plot(x2, y_rec_6_prc, label=labels[6], marker='s')
plt.plot(x2, y_rec_7_prc, label=labels[7], marker='*')
plt.plot(x2, y_rec_8_prc, label=labels[8], marker='p')
plt.xticks(x2, x)
plt.legend(bbox_to_anchor=(1.04, 1))
plt.ylabel('Valor')
plt.xlabel('Tamanho da lista de recomendação')
plt.show()
```



In [11]:
```
y_rec_1_ap = [recomendations_1.ap_3.mean(), recomendations_1.ap_5.mean(), recome
y_rec_2_ap = [recomendations_2.ap_3.mean(), recomendations_2.ap_5.mean(), recome
y_rec_3_ap = [recomendations_3.ap_3.mean(), recomendations_3.ap_5.mean(), recome
y_rec_4_ap = [recomendations_4.ap_3.mean(), recomendations_4.ap_5.mean(), recome
y_rec_5_ap = [recomendations_5.ap_3.mean(), recomendations_5.ap_5.mean(), recome
y_rec_6_ap = [recomendations_6.ap_3.mean(), recomendations_6.ap_5.mean(), recome
y_rec_7_ap = [recomendations_7.ap_3.mean(), recomendations_7.ap_5.mean(), recome
y_rec_8_ap = [recomendations_8.ap_3.mean(), recomendations_8.ap_5.mean(), recome
```

In [55]:
```
((recomendations_4.ap_5.mean() / recomendations_1.ap_5.mean()) - 1) * 100
```

Out[55]:
```
0.7972020778686639
```

In [47]:
```

```

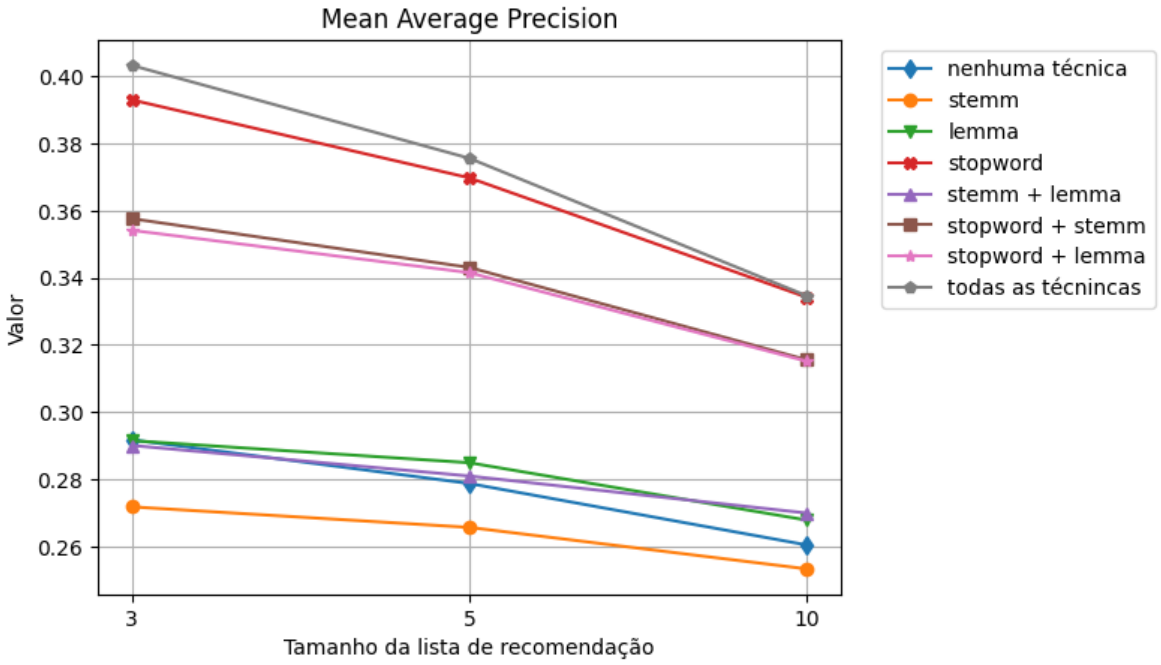
Out[47]:
```
0.3541218637992831
```

In [61]:
```
min_aum_3 = (((recomendations_7.ap_3.mean() / recomendations_1.ap_3.mean()) - 1)
max_aum_3 = (((recomendations_8.ap_3.mean() / recomendations_1.ap_3.mean()) - 1)
min_aum_5 = (((recomendations_4.ap_5.mean() / recomendations_1.ap_5.mean()) - 1)
max_aum_5 = (((recomendations_8.ap_5.mean() / recomendations_1.ap_5.mean()) - 1)
min_aum_10 = (((recomendations_3.ap_10.mean() / recomendations_1.ap_10.mean()) -
max_aum_10 = (((recomendations_8.ap_10.mean() / recomendations_1.ap_10.mean()) -
```

In [62]:
```
print(f'Tamanho de lista 3: aumento de {min_aum_3:.2f}% a {max_aum_3:.2f}%')
print(f'Tamanho de lista 5: aumento de {min_aum_5:.2f}% a {max_aum_5:.2f}%')
print(f'Tamanho de lista 10: aumento de {min_aum_10:.2f}% a {max_aum_10:.2f}%')
```

```
Tamanho de lista 3: aumento de 21.38% a 38.21%
Tamanho de lista 5: aumento de 0.80% a 34.72%
Tamanho de lista 10: aumento de 2.84% a 28.47%
```

In [12]:
```
plt.title('Mean Average Precision')
plt.grid()
plt.plot(x2, y_rec_1_ap, label=labels[1], marker='d')
plt.plot(x2, y_rec_2_ap, label=labels[2], marker='o')
plt.plot(x2, y_rec_3_ap, label=labels[3], marker='v')
plt.plot(x2, y_rec_5_ap, label=labels[5], marker='X')
plt.plot(x2, y_rec_4_ap, label=labels[4], marker='^')
plt.plot(x2, y_rec_6_ap, label=labels[6], marker='s')
plt.plot(x2, y_rec_7_ap, label=labels[7], marker='*')
plt.plot(x2, y_rec_8_ap, label=labels[8], marker='p')
plt.xticks(x2, x)
plt.legend(bbox_to_anchor=(1.04, 1))
plt.ylabel('Valor')
plt.xlabel('Tamanho da lista de recomendação')
plt.show()
```



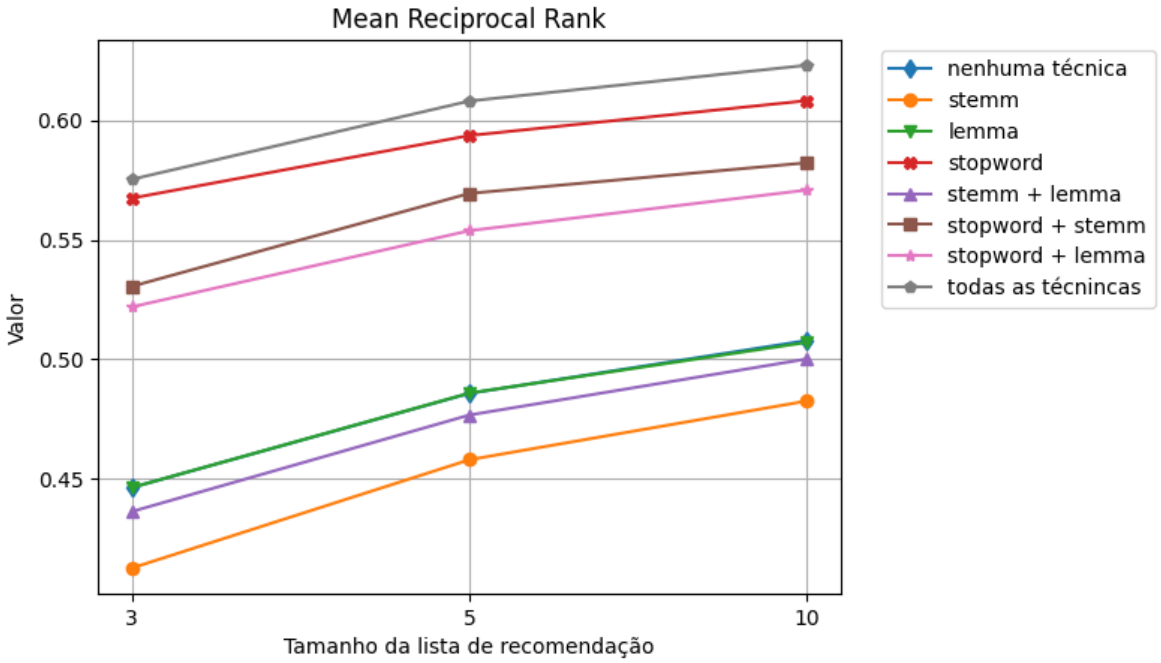
In [13]:
```
y_rec_1_rr = [recomendations_1.rr_3.mean(), recomendations_1.rr_5.mean(), recome
y_rec_2_rr = [recomendations_2.rr_3.mean(), recomendations_2.rr_5.mean(), recome
y_rec_3_rr = [recomendations_3.rr_3.mean(), recomendations_3.rr_5.mean(), recome
y_rec_4_rr = [recomendations_4.rr_3.mean(), recomendations_4.rr_5.mean(), recome
y_rec_5_rr = [recomendations_5.rr_3.mean(), recomendations_5.rr_5.mean(), recome
y_rec_6_rr = [recomendations_6.rr_3.mean(), recomendations_6.rr_5.mean(), recome
y_rec_7_rr = [recomendations_7.rr_3.mean(), recomendations_7.rr_5.mean(), recome
y_rec_8_rr = [recomendations_8.rr_3.mean(), recomendations_8.rr_5.mean(), recome
```

In [46]:
```
print(recomendations_1.rr_3.mean())
print(recomendations_7.rr_3.mean())
```

```
0.44623655913978494
0.5218637992831541
```

In [15]:
```
plt.title('Mean Reciprocal Rank')
plt.grid()
plt.plot(x2, y_rec_1_rr, label=labels[1], marker='d')
plt.plot(x2, y_rec_2_rr, label=labels[2], marker='o')
```

```
plt.plot(x2, y_rec_3_rr, label=labels[3], marker='v')
plt.plot(x2, y_rec_5_rr, label=labels[5], marker='X')
plt.plot(x2, y_rec_4_rr, label=labels[4], marker='^')
plt.plot(x2, y_rec_6_rr, label=labels[6], marker='s')
plt.plot(x2, y_rec_7_rr, label=labels[7], marker='*')
plt.plot(x2, y_rec_8_rr, label=labels[8], marker='p')
plt.xticks(x2, x)
plt.legend(bbox_to_anchor=(1.04, 1))
plt.ylabel('Valor')
plt.xlabel('Tamanho da lista de recomendação')
plt.show()
```



In [72]:
```
recomendations_1.rr_5.mean()
```

Out[72]:
```
0.48569892473118287
```

In [71]:
```
recomendations_3.rr_5.mean()
```

Out[71]:
```
0.48580645161290326
```

In [73]:
```
min_aum_3 = (((recomendations_7.rr_3.mean() / recomendations_1.rr_3.mean()) - 1)
max_aum_3 = (((recomendations_8.rr_3.mean() / recomendations_1.rr_3.mean()) - 1)
min_aum_5 = (((recomendations_7.rr_5.mean() / recomendations_1.rr_5.mean()) - 1)
max_aum_5 = (((recomendations_8.rr_5.mean() / recomendations_1.rr_5.mean()) - 1)
min_aum_10 = (((recomendations_7.rr_10.mean() / recomendations_1.rr_10.mean()) -
max_aum_10 = (((recomendations_8.rr_10.mean() / recomendations_1.rr_10.mean()) -
```

In [74]:
```
print(f'Tamanho de lista 3: aumento de {min_aum_3:.2f}% a {max_aum_3:.2f}%')
print(f'Tamanho de lista 5: aumento de {min_aum_5:.2f}% a {max_aum_5:.2f}%')
print(f'Tamanho de lista 10: aumento de {min_aum_10:.2f}% a {max_aum_10:.2f}%')
```

```
Tamanho de lista 3: aumento de 16.95% a 28.92%
Tamanho de lista 5: aumento de 14.02% a 25.19%
Tamanho de lista 10: aumento de 12.40% a 22.69%
```