LDA Topic Model Classifier

In [56]:
```
from pickle import load#, dump
from my_lda_utils import my_lda_model
import pandas as pd
import numpy as np
from nltk.stem import WordNetLemmatizer, SnowballStemmer
import seaborn as sns
import matplotlib.pyplot as plt
```

Import Data

In [3]:
```
clean_data_folder = "../clean_data/"
with open("../clean_data/"+"characters_with_gender.pckl", "rb") as f2:
    characters = load(f2)
```

In [4]:
```
clean_data_folder_lda = "../clean_data_lda"
with open(clean_data_folder_lda+"/all_data.pkl", "rb") as f:
    all_scripts = load(f)
data_lda = pd.merge(all_scripts, characters, on="character", how="inner")
```

In [5]:
```
all_scripts.reset_index(inplace=True, drop=True)
```

In [6]:
```
data = pd.merge(all_scripts, characters, on="character", how="inner")
```

In [7]:
```
counts = data.gender2.value_counts()
```

In [8]:
```
data
```

Out[8]:

	character	text	season	episode	doc_id	gender	gender2
0	woman	[are, not, you, guys, afraid, of, an, officer,...	1	1	3	female	female
1	woman	[does, that, thing, always, do, that]	1	1	5	female	female
2	woman	[whatever, it, is, under, the, trap, i, just, ...	1	1	6	female	female
3	woman	[im, telling, you, that, thing, is, moving]	1	1	8	female	female
4	woman	[i, take, it, this, has, never, happened, before]	1	1	11	female	female
...	...	...	...	...	...	...	...
40859	volnek	[you, shot, me]	9	8	34108	male	male
40860	volnek	[i, will, have, vengeance]	9	8	34108	male	male
40861	volnek	[you, are, fortunate, my, brother, drugged, yo...	9	8	34109	male	male
40862	volnek	[you, were, lucky]	9	8	34109	male	male
40863	volnek	[gate, kawooshes, granting, my, freedom, chang...	9	8	34109	male	male

40864 rows × 7 columns

Gender count stats for later

In [11]:
```
#gender counted by character occurrence
counts = data.gender2.value_counts()

female_occurr = counts.female
male_occurr = counts.male
unclear_occurr = counts.unclear
occourence_count = female_occurr+male_occurr+unclear_occurr


#gender counted by character count
data1 = data[["character", "gender2"]].drop_duplicates()
data2 = data1.groupby(["gender2"]).count()
data3 =data2.reset_index()

female_count = data3.loc[data3.gender2 == "female"].character[0]
male_count = data3.loc[data3.gender2 == "male"].character[1]
unclear_count = data3.loc[data3.gender2 == "unclear"].character[2]
character_count = female_count+male_count+unclear_count
```

Transform data

In [12]:
```
data
```

Out[12]:

	character	text	season	episode	doc_id	gender	gender2
0	woman	[are, not, you, guys, afraid, of, an, officer,...	1	1	3	female	female
1	woman	[does, that, thing, always, do, that]	1	1	5	female	female
2	woman	[whatever, it, is, under, the, trap, i, just, ...	1	1	6	female	female
3	woman	[im, telling, you, that, thing, is, moving]	1	1	8	female	female
4	woman	[i, take, it, this, has, never, happened, before]	1	1	11	female	female
...	...	...	...	...	...	...	...
40859	volnek	[you, shot, me]	9	8	34108	male	male
40860	volnek	[i, will, have, vengeance]	9	8	34108	male	male
40861	volnek	[you, are, fortunate, my, brother, drugged, yo...	9	8	34109	male	male
40862	volnek	[you, were, lucky]	9	8	34109	male	male
40863	volnek	[gate, kawooshes, granting, my, freedom, chang...	9	8	34109	male	male

40864 rows × 7 columns

In [13]:
```
data_rm_gender1 = data.drop(['gender'], axis=1)
```

In [14]:
```
data_rename_col = data_rm_gender1.rename(columns={"gender2": "gender", "doc_id": "scene_id"}, errors="raise")
```

In [15]:
```
data_groupped = data_rename_col.groupby(["character", "season", "episode", "scene_id", "gender"]).agg(sum)
```

In [16]:
```
data_reset_index = data_groupped.reset_index()
```

In [17]:
```
data_reset_index.head()
```

Out[17]:

	character	season	episode	scene_id	gender	text
0	.	1	3	2079	unclear	[sam, is, with, abu, they, are, sitting, among...
1	.	1	16	1164	unclear	[]
2	.	1	17	1324	unclear	[]
3	.	2	9	6367	unclear	[]
4	.	2	10	3304	unclear	[]

In [18]:
```
# remove empty texts
data = data_reset_index.loc[data_reset_index.text.map(len) > 0]
data.head()
```

Out[18]:

	character	season	episode	scene_id	gender	text
0	.	1	3	2079	unclear	[sam, is, with, abu, they, are, sitting, among...
9	.	2	20	4835	unclear	[lockdown, in, progress, stand, clear, of, all...
13	bigwig	5	12	14914	male	[look, i, know, it, says, hes, flightless, in,...
14	bigwig	5	12	14915	male	[who, is, this]
15	bigwig	5	12	14916	male	[you, are, telling, me, an, air, force, office...

Classify

In [19]:
```
def lda_calssify(lda_model, dictionary, text):
    # create corpus
    bow = dictionary.doc2bow(text)
    topics = (lda_model.get_document_topics(bow,
                                            minimum_probability=0.0))
    #get topic with highes probability
    highest_percentage = 0
    classified_topic = 0
    for topic in topics:
        if topic[1] > highest_percentage:
            highest_percentage = topic[1]
            classified_topic = topic[0]
    return(classified_topic)
```

In [20]:
```
#get a classifying model:
with open("tfidf_coherence-topics-5-passes-5_approach3.pkl", "rb") as f:
    df = load(f)
my_lda_model_1 = df.model[0.01]
my_lda_model_4 = df.model[0.04]
my_lda_model_8 = df.model[0.08]
```

In [21]:
```
df_columns = list(data.columns)
df_columns.append("topic")
```

In [22]:
```
data
```

Out[22]:

	character	season	episode	scene_id	gender	text
0	.	1	3	2079	unclear	[sam, is, with, abu, they, are, sitting, among...
9	.	2	20	4835	unclear	[lockdown, in, progress, stand, clear, of, all...
13	bigwig	5	12	14914	male	[look, i, know, it, says, hes, flightless, in,...
14	bigwig	5	12	14915	male	[who, is, this]
15	bigwig	5	12	14916	male	[you, are, telling, me, an, air, force, office...
...	...	...	...	...	...	...
26344	zippy	3	15	7971	male	[then, i, wish, to, point, out, the, futility,...
26345	zippy	3	15	7981	male	[my, vessel, comes, in, anticipation, of, our,...
26346	zippy	3	15	7982	male	[the, goals, rest, our, case, and, we, are, pr...
26347	zippy	3	15	7993	male	[stands, we, are, in, favor, of, korea, sits]
26348	zippy	3	15	8002	male	[talking, into, ball, rita, retook, data, i, d...

26249 rows × 6 columns

In [23]:
```
# Classify data with model 1
df1_1 = data.apply(lambda x: [x[0], x[1], x[2], x[3], x[4], x[5],
                              lda_calssify(my_lda_model_1.lda_model, my_lda_model_1.dictionary, x[5])], axis=1, result_type='expand')

# Classify data with model 4
df1_4 = data.apply(lambda x: [x[0], x[1], x[2], x[3], x[4], x[5],
                              lda_calssify(my_lda_model_4.lda_model, my_lda_model_4.dictionary, x[5])], axis=1, result_type='expand')
# Classify data with model 8
df1_8 = data.apply(lambda x: [x[0], x[1], x[2], x[3], x[4], x[5],
                              lda_calssify(my_lda_model_8.lda_model, my_lda_model_8.dictionary, x[5])], axis=1, result_type='expand')
```

In [24]:
```
df1_1.columns = df1_4.columns = df1_8.columns = df_columns
df1_1.head()
```

Out[24]:

	character	season	episode	scene_id	gender	text	topic
0	.	1	3	2079	unclear	[sam, is, with, abu, they, are, sitting, among...	2
9	.	2	20	4835	unclear	[lockdown, in, progress, stand, clear, of, all...	4
13	bigwig	5	12	14914	male	[look, i, know, it, says, hes, flightless, in,...	1
14	bigwig	5	12	14915	male	[who, is, this]	0
15	bigwig	5	12	14916	male	[you, are, telling, me, an, air, force, office...	1

In [25]:
```
occournce = df1_1.gender.value_counts()
```

Plot

Model 1

In [26]:
```
df2 = df1_1[["gender", "topic"]]
df2["count"] = 1
```

```
C:\Users\debor\AppData\Local\Temp\ipykernel_2228\2464217104.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df2["count"] = 1
```

In [27]:
```
df3 = df2.groupby(["gender", "topic"]).count()
```

In [28]:
```
df4 =df3.reset_index()
```

In [29]:
```
df5 = df4.pivot_table("count", ['topic'], 'gender')
df5
```

Out[29]:

gender	female	male	unclear
topic			
0	2946	8264	130
1	927	2654	59
2	1163	2705	59
3	1110	3029	70
4	866	2220	47

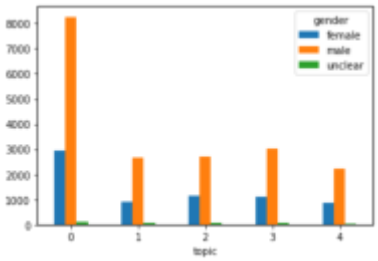
In [30]:
```
df6 = df5.copy()
df7 = df5.copy()
```

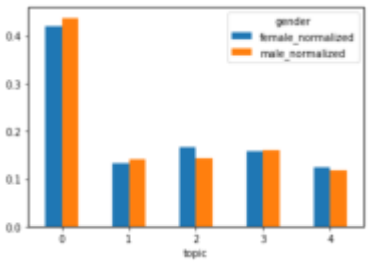
In [31]:
```
# Normalize by occourence count
df6["female_normalized"] = df5.female/occournce.female
df6["male_normalized"] = df5.male/occournce.male
df6["unclear_normalized"] = df5.unclear/occournce.unclear
#df6 =df6.drop(["female", "male", "unclear"], axis=1)
df6
```

Out[31]:

gender	female	male	unclear	female_normalized	male_normalized	unclear_normalized
topic						
0	2946	8264	130	0.420137	0.437897	0.356164
1	927	2654	59	0.132202	0.140632	0.161644
2	1163	2705	59	0.165859	0.143334	0.161644
3	1110	3029	70	0.158300	0.160502	0.191781
4	866	2220	47	0.123503	0.117635	0.128767

In [32]:
```
ax = df5.plot.bar(rot=0)
ax = df6[["female_normalized", "male_normalized"]].plot.bar(rot=0)
```





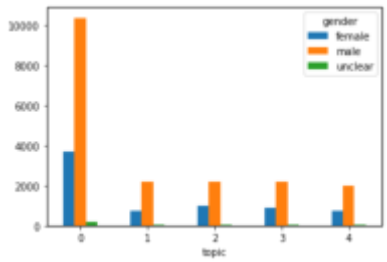
In [33]:
```
df6['avg'] = (df6.female_normalized+df6.male_normalized)/2
df6['dev_f'] = np.power((df6.avg-df6.female_normalized), 2)
df6['dev_m'] = np.power((df6.avg-df6.male_normalized), 2)
df6['std_dev'] = (df6['dev_f']+df6['dev_m'])/2
format(df6.std_dev.mean(), '.8f')
```

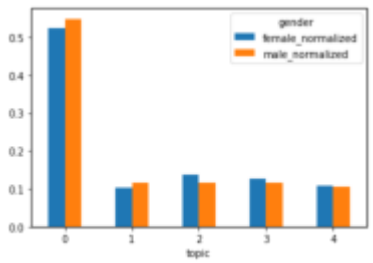
Out[33]:
```
'0.00004666'
```

Model 4

In [34]:
```
df2 = df1_4[["gender", "topic"]]
df2["count"] = 1
df3 = df2.groupby(["gender", "topic"]).count()
df4 =df3.reset_index()
df5 = df4.pivot_table("count", ['topic'], 'gender')
df6 = df5.copy()
df7 = df5.copy()
# Normalize by occourence count
df6["female_normalized"] = df5.female/occournce.female
df6["male_normalized"] = df5.male/occournce.male
df6["unclear_normalized"] = df5.unclear/occournce.unclear
df6 =df6.drop(["female", "male", "unclear"], axis=1)
ax = df5.plot.bar(rot=0)
ax = df6[["female normalized", "male normalized"]].plot.bar(rot=0)
```

```
C:\Users\debor\AppData\Local\Temp\ipykernel_2228\976086660.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df2["count"] = 1
```





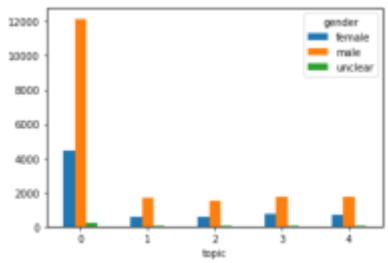
In [35]:
```
df6['avg'] = (df6.female_normalized+df6.male_normalized)/2
df6['dev_f'] = np.power((df6.avg-df6.female_normalized), 2)
df6['dev_m'] = np.power((df6.avg-df6.male_normalized), 2)
df6['std_dev'] = (df6['dev_f']+df6['dev_m'])/2
format(df6.std_dev.mean(), '.8f')
```

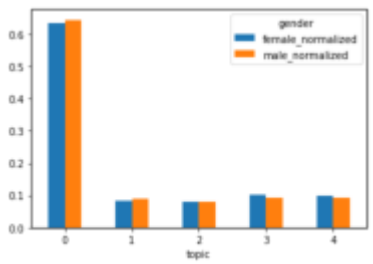
Out[35]: '0.00006703'

Model 8

In [36]:
```
df2 = df1_8[["gender", "topic"]]
df2["count"] = 1
df3 = df2.groupby(["gender", "topic"]).count()
df4 =df3.reset_index()
df5 = df4.pivot_table("count", ['topic'], 'gender')
df6 = df5.copy()
df7 = df5.copy()
# Normalize by occourence count
df6["female_normalized"] = df5.female/occournce.female
df6["male_normalized"] = df5.male/occournce.male
df6["unclear_normalized"] = df5.unclear/occournce.unclear
df6 =df6.drop(["female", "male", "unclear"], axis=1)
ax = df5.plot.bar(rot=0)
ax = df6[["female normalized", "male normalized"]].plot.bar(rot=0)
```

```
C:\Users\debor\AppData\Local\Temp\ipykernel_2228\1371994753.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df2["count"] = 1
```





In [97]:
```
df8 = df6.copy()
df9 = df8.rename(columns={"female_normalized": "Female",
                          "male_normalized": "Male"})
df9.columns.name = "Gender"
df9.index.name = "Topic"
ax = df9[["Female", "Male"]].plot.bar(rot=0,
                                      ylabel="Topic percentage",
                                     title="Topic Percentage in All Text of Genders Male and Female")

```

In [95]:
```
df6
```

Out[95]:

gender	female_normalized	male_normalized	unclear_normalized	avg	dev_f	dev_m	std_dev
topic							
0	0.633485	0.644235	0.567123	0.638860	2.888736e-05	2.888736e-05	2.888736e-05
1	0.084284	0.089922	0.106849	0.087103	7.945330e-06	7.945330e-06	7.945330e-06
2	0.081004	0.080702	0.084932	0.080853	2.286517e-08	2.286517e-08	2.286517e-08
3	0.103394	0.091670	0.109589	0.097532	3.436294e-05	3.436294e-05	3.436294e-05
4	0.097832	0.093472	0.131507	0.095652	4.753441e-06	4.753441e-06	4.753441e-06

In [88]:
```
df9
```

Out[88]:

Gender	Female	Male	unclear_normalized	avg	dev_f	dev_m	std_dev
Topic							
0	0.633485	0.644235	0.567123	0.638860	2.888736e-05	2.888736e-05	2.888736e-05
1	0.084284	0.089922	0.106849	0.087103	7.945330e-06	7.945330e-06	7.945330e-06
2	0.081004	0.080702	0.084932	0.080853	2.286517e-08	2.286517e-08	2.286517e-08
3	0.103394	0.091670	0.109589	0.097532	3.436294e-05	3.436294e-05	3.436294e-05
4	0.097832	0.093472	0.131507	0.095652	4.753441e-06	4.753441e-06	4.753441e-06

In [37]:
```
df6['avg'] = (df6.female_normalized+df6.male_normalized)/2
df6['dev_f'] = np.power((df6.avg-df6.female_normalized), 2)
df6['dev_m'] = np.power((df6.avg-df6.male_normalized), 2)
df6['std_dev'] = (df6['dev_f']+df6['dev_m'])/2
format(df6.std_dev.mean(), '.8f')
```

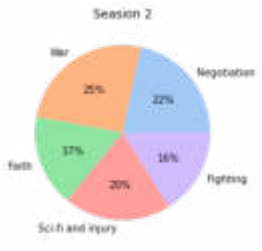
Out[37]: '0.00001519'

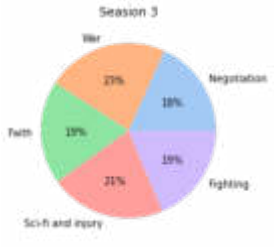
Inspect Seasons

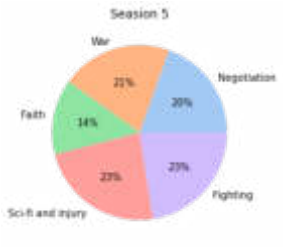
In [69]:
```
def show_topics(scripts, i):

    #extract the texts and group them together
    texts = sum(list(scripts["text"]), [])

    # get the topic model and dictionary
    with open ("lda_topic_model_-5_topics-5_passes-0.08_tfidf_threshold.pkl", "rb") as f:
        my_lda_topic_model = load(f)
    model = my_lda_topic_model.lda_model
    dictionary = my_lda_topic_model.dictionary

    # get the topics
    bow = dictionary.doc2bow(texts)
    topics ={0:"Negotiation",
             1: "War",
             2: "Faith",
             3: "Sci-fi and injury",
             4: "Fighting"}

    text_topics = (model.get_document_topics(bow,
                                             minimum_probability=0.0))
    df = pd.DataFrame(text_topics)
    df =df.replace(topics)
    df.columns = ["Topic", "Percentage"]
   # df =df.set_index("Topic")


    labels = topics.values
    colors = sns.color_palette('pastel')[0:5]
    plt.pie(df.Percentage, labels=df.Topic, colors = colors, autopct='%.0f%%')
    plt.title("Seasion "+str(i))
    plt.show()
    return
```

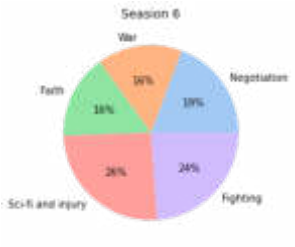
In [70]:

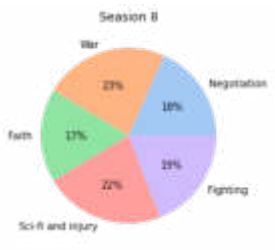
```
for i in range(1,10):
    scripts = data[data.season == i]
    show_topics(scripts, i)

```











In [62]:
```
1 data[data.season == 3]
```

Out[62]:

	character	season	episode	scene_id	gender	text
40	boy	3	5	9913	male	[colonel, jack, what, are, you, doing, here]
1919	daniel_jackson	3	1	6510	male	[i, really, try, not, to]
1920	daniel_jackson	3	1	6578	male	[wait, what, about, jack]
1921	daniel_jackson	3	1	6605	male	[its, just, a, deep, bleeding, gas, but, till,...
1922	daniel_jackson	3	1	6614	male	[well, i, guess, we, ca, not, go, under, it, e...
...	...	...	...	...	...	...
26344	zippy	3	15	7971	male	[then, i, wish, to, point, out, the, futility,...
26345	zippy	3	15	7981	male	[my, vessel, comes, in, anticipation, of, our,...
26346	zippy	3	15	7982	male	[the, goals, rest, our, case, and, we, are, pr...
26347	zippy	3	15	7993	male	[stands, we, are, in, favor, of, korea, sits]
26348	zippy	3	15	8002	male	[talking, into, ball, rita, retook, data, i, d...

3424 rows × 6 columns