Overview Generation

In [26]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
from pickle import load, dump
import numpy as np
from afinn import Afinn
import warnings
warnings.filterwarnings('ignore')
import pyLDAvis
pyLDAvis.enable_notebook()
from my_lda_utils import my_lda_model
```

In [14]:
```
def get_all_scripts():
    clean_data_folder = "../clean_data"
    filenames = os.listdir(clean_data_folder)
    os.listdir(clean_data_folder)

    with open(clean_data_folder+"/all_data.pkl", "rb") as f:
        all_scripts = load(f)
    all_scripts['episode_str']=all_scripts['episode'].apply(lambda x: '0'+str(x) if len(str(x))==1 else str(x))
    all_scripts['season_episode_no'] = all_scripts['season'].apply(lambda x: str(x))+all_scripts['episode_str']
    all_scripts['season_episode_no'] = all_scripts['season_episode_no'].apply(lambda x: int(x))
    return all_scripts
```

In [15]:
```
def extract_specific_episode(all_scripts, season_no, episode_no):
    episode_script=all_scripts[all_scripts.season==season_no]
    episode_script=all_scripts[all_scripts.episode==episode_no]

    if episode_script.empty: #saveguard if episode or season does not exist
        print("Season or episode does not exist.")
        return
    return episode_script
```

In [16]:
```
def list_charaters(script):
    characters = script["character"].unique()

    print(f"There are {len(characters)} characters in the season.")
    text ="These characters are "
    i=0
    for name in characters:
        if i + 1 == len(characters):
            text= text +"and "+ name + "."
        else:
            text= text + name + ", "
        i=i+1

    print(text+"\n\n")
    return
```

In [17]:
```
def show_three_characters_with_biggest_amount_of_speech(script):
    characters_word_count = script[["character", "word_count"]].groupby("character",as_index=False).sum()
    characters_word_count=characters_word_count.sort_values(by='word_count', ascending=False)
    characters_word_count=characters_word_count.reset_index(drop=True)

    top_3 = characters_word_count['character'].head(3).values.tolist()
    top_3_text ="The 3 with the highest word counts are "
    i=0
    for name in top_3:
        if i + 1 == len(top_3):
            top_3_text= top_3_text +"and "+ name + "."
        else:
            top_3_text= top_3_text + name + ", "
        i=i+1

    print(top_3_text+"\n\n")
    return top_3
```

In [18]:
```
def show_amount_of_dialogue(script):
    dialogue= script.word_count.sum()
    dialogue_text= "In total, there are "+str(dialogue)+" words spoken."
    print(dialogue_text)
    return
```

In [19]:
```
def sentiment(script):

    afn = Afinn(emoticons=True)
    afinn_wl_url = ('https://raw.githubusercontent.com'
                    '/fnielsen/afinn/master/afinn/data/AFINN-111.txt')
    afinn_wl_df = pd.read_csv(afinn_wl_url,
                              header=None, # no column names
                              sep='\t',  # tab sepeated
                              names=['term', 'value']) #new column names
    seed = 808 # seed for sample so results are stable
    afinn_wl_df.sample(15, random_state = seed)

    script['text']=script.text.apply(lambda x: ' '.join(x))
    script['sentiment_score']=script.text.apply(lambda x: afn.score(x))
    script=script.reset_index(drop=True)
    avg_sentiment=script.sentiment_score.mean()

    print(f"The sentiment score over the episode/season has an average of {round(avg_sentiment,4)}.\nThe graph below shows the sentiment for each line in chronological order.")
    g = sns.relplot(
    data=script,
    x=script.index, y="sentiment_score", kind="line",
    height=5, aspect=5, color="blue"
    ).set(
        title="Sentiments over the episode or season",
        ylabel="sentiment score",
        xlabel="line in script"
    )
    g.despine(left=True)

    ax1, = g.axes[0]

    ax1.axhline(avg_sentiment, ls='--', c="red")
    plt.legend(labels=["Sentiment","Average"])
    plt.xlim([0, script.shape[0]])
    plt.show()
    return
```

In [20]:
```
def timeline_with_amount_of_speech_for_top3_characters(season_script, top_3):
    print("The amount of speech each of the top 3 characters has per episode, is as follows:")

    amount_speech_top1 = season_script[season_script.character==top_3[0]].groupby("episode").sum().reset_index()
    amount_speech_top1=amount_speech_top1[["episode","word_count"]]
    amount_speech_top1=amount_speech_top1.rename(columns={"word_count": top_3[0]})
    amount_speech_top1['episode'] = amount_speech_top1['episode'].apply(lambda x: str(x))

    amount_speech_top2 = season_script[season_script.character==top_3[1]].groupby("episode").sum().reset_index()
    amount_speech_top2=amount_speech_top2[["episode","word_count"]]
    amount_speech_top2=amount_speech_top2.rename(columns={"word_count": top_3[1]})
    amount_speech_top2['episode'] = amount_speech_top2['episode'].apply(lambda x: str(x))

    amount_speech_top3 = season_script[season_script.character==top_3[2]].groupby("episode").sum().reset_index()
    amount_speech_top3=amount_speech_top3[["episode","word_count"]]
    amount_speech_top3=amount_speech_top3.rename(columns={"word_count": top_3[2]})
    amount_speech_top3['episode'] = amount_speech_top3['episode'].apply(lambda x: str(x))

    amount_speech_top1[top_3[1]]=amount_speech_top2[top_3[1]]
    amount_speech_top1[top_3[2]]=amount_speech_top3[top_3[2]]

    amount_speech_top1.plot(x='episode', y=[top_3[0],top_3[1],top_3[2]], figsize=(21,5), xlabel='Episode', ylabel='Word count', title='Word count of the top 3 characters per episode');
    plt.show()
    return
```

In [21]:
```
def show_topics(scripts):

    #extract the texts and group them together
    texts = list(scripts["text"])#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("Topic Percentage")
    plt.show()
    return
```

In [22]:
```
def generate_overview_season(season_no):

    all_scripts= get_all_scripts()
    season_script=all_scripts[all_scripts.season==season_no]

    if season_script.empty: #saveguard if episode or season does not exist
        print("Season does not exist.")
        return
    print('\033[1m'+"Shown is an overview over season ", season_no, "\n")
    print('\033[0m')

    list_charaters(season_script)
    top_3= show_three_characters_with_biggest_amount_of_speech(season_script)
    show_amount_of_dialogue(season_script)
    sentiment(season_script)
    timeline_with_amount_of_speech_for_top3_characters(season_script, top_3)
    show_topics(season_script)
    return
```

In [23]:
```
def generate_overview_episode(season_no,episode_no):

    all_scripts= get_all_scripts()
    episode_script=all_scripts[(all_scripts.season==season_no) & (all_scripts.episode==episode_no)]
#     episode_script=episode_script[all_scripts.episode==episode_no]

    if episode_script.empty: #saveguard if episode or season does not exist
        print("Season or episode does not exist.")
        return
    print('\033[1m'+"Shown is an overview over season ", season_no, " episode ", episode_no,"\n")
    print('\033[0m')

    list_charaters(episode_script)
    top_3= show_three_characters_with_biggest_amount_of_speech(episode_script)
    show_amount_of_dialogue(episode_script)
    sentiment(episode_script)
    show_topics(episode_script)
    return
```

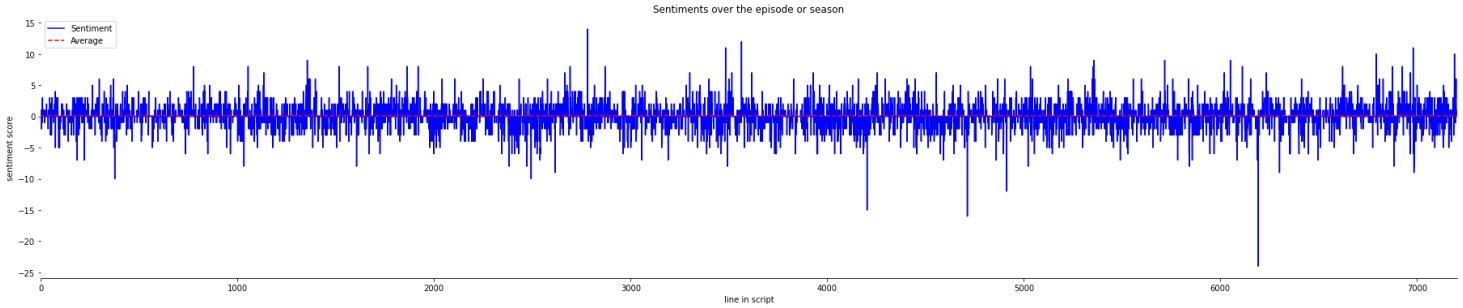
In [24]:
```
generate_overview_season(1)
```

Shown is an overview over season 1

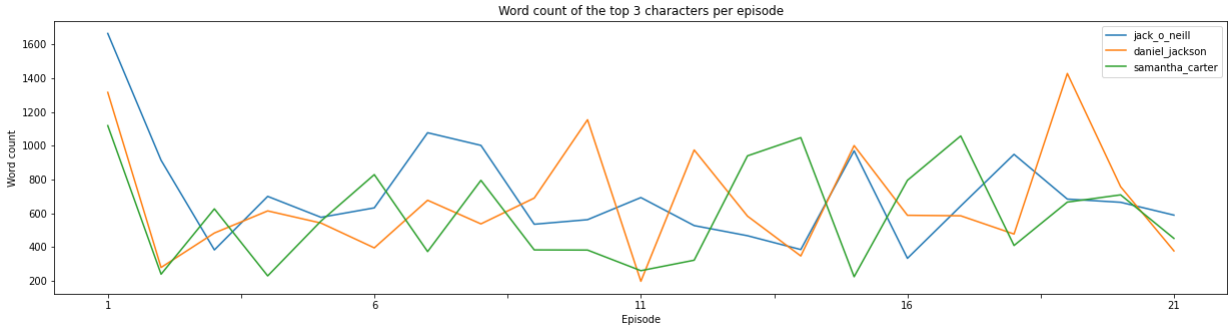
There are 106 characters in the season.
These characters are airman, woman, officer, apophis, soldier, samantha_carter, jack_o_neill, hammond, warner, kawalsky, harriman, ferretti, daniel_jackson, skaara, sha're, teal_c, bo'la, boy, tech, medic, warren, native, goa'uld, casey, dr langford, scientist, catherine, martha, earnest, dr_janet, priest, bra'tac, o`neill, drey'auc, rya'c, nem, mackensie, kleinhouse, cole, hathor, solder, cassie, nurse, dr warner, davis, hanno, man, young hanno, villager, shak'l, narim, omoc, tuplo, maybourne, lya, siler, harlan, tv reporter, soldier 2, solder 3, airmen, jaffa, "auto destruct in, walter, doctor, kennedy, kinsey, entity, voice, ml, klorel, abu, ., mughal, turghan, nya, guard, makepeace, marine, leedora, johnson, connor, franks, baker, hanson, jamala, sara, dad, reporter, sara's dad, crystal, answering machine, police officer, charlie, secretary, nefrayu, oper, anteaus, alekos, thetyes, kynthia, argosian, gairwyn, thor, kendra, and unas.

The 3 with the highest word counts are jack_o_neill, daniel_jackson, and samantha_carter.

In total, there are 76413 words spoken.
The sentiment score over the episode/season has an average of 0.0303.
The graph below shows the sentiment for each line in chronological order.



The amount of speech each of the top 3 characters has per episode, is as follows:



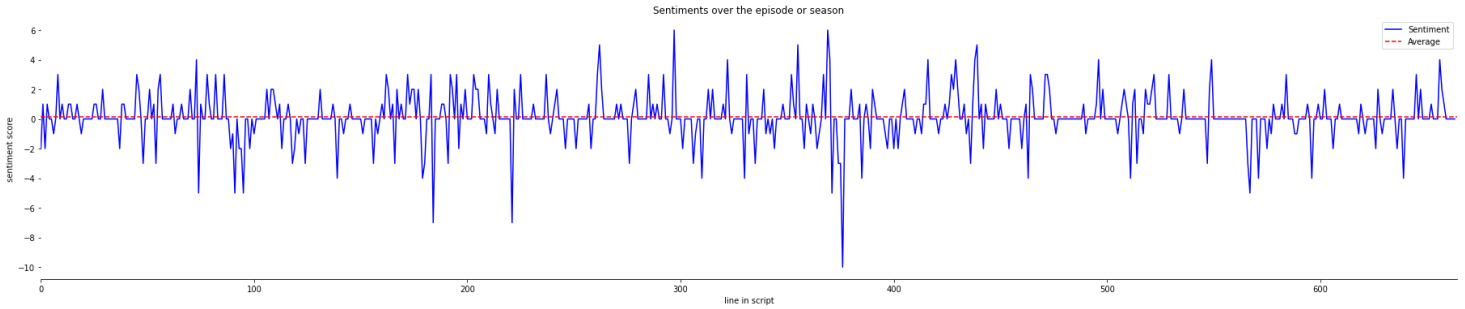
In [25]:
```
generate_overview_episode(1,1)
```

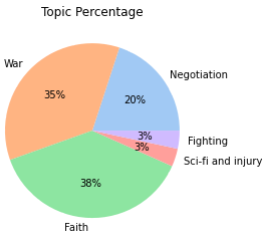
Shown is an overview over season 1 episode 1

There are 24 characters in the season.
These characters are airman, woman, officer, apophis, soldier, samantha_carter, jack_o_neill, hammond, warner, kawalsky, harriman, ferretti, daniel_jackson, skaara, sha're, teal_c, bo'la, boy, tech, medic, warren, native, goa'uld, and casey.

The 3 with the highest word counts are jack_o_neill, daniel_jackson, and samantha_carter.

In total, there are 6472 words spoken.
The sentiment score over the episode/season has an average of 0.1205.
The graph below shows the sentiment for each line in chronological order.





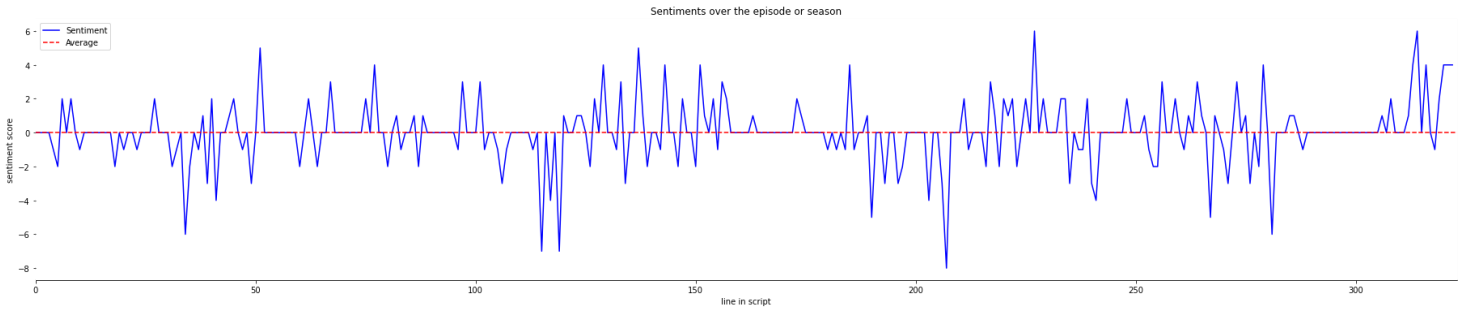
In [27]:
```
generate_overview_episode(3,19)
```

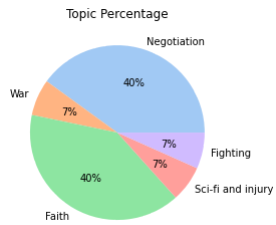
Shown is an overview over season 3 episode 19

There are 15 characters in the season.
These characters are technician, hammond, samantha_carter, woman, nyan, daniel_jackson, jack_o_neill, mallen, teal_c, limp, guard, unw, rigar, guy, and dr_janet.

The 3 with the highest word counts are nyan, rigar, and teal_c.

In total, there are 3081 words spoken.
The sentiment score over the episode/season has an average of 0.0186.
The graph below shows the sentiment for each line in chronological order.





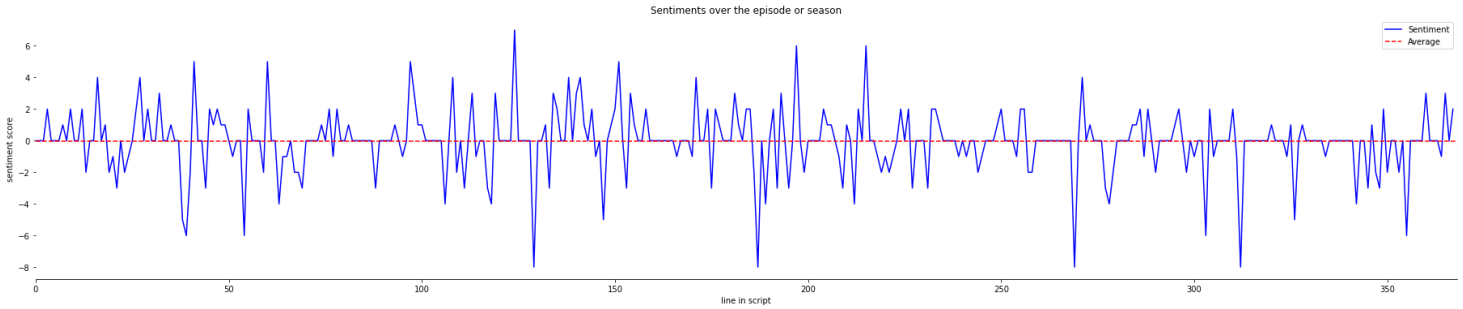
In [28]:
```
generate_overview_episode(6,7)
```

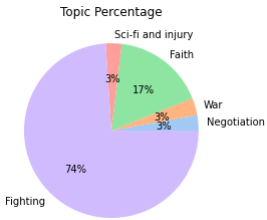
Shown is an overview over season 6 episode 7

There are 22 characters in the season.
These characters are sgt. davis, hammond, samantha_carter, voice over speaker, briefing room, jonas, jack_o_neill, teal_c, 'gateroom, hale, kieran, dreylock, control room, kelowna, valis, resistance leader, kelownan conference chamber, dr. kieran's lab, woman, sgc infirmary, dr_janet, and sgc briefing room.

The 3 with the highest word counts are jonas, kieran, and valis.

In total, there are 5084 words spoken.
The sentiment score over the episode/season has an average of -0.0489.
The graph below shows the sentiment for each line in chronological order.





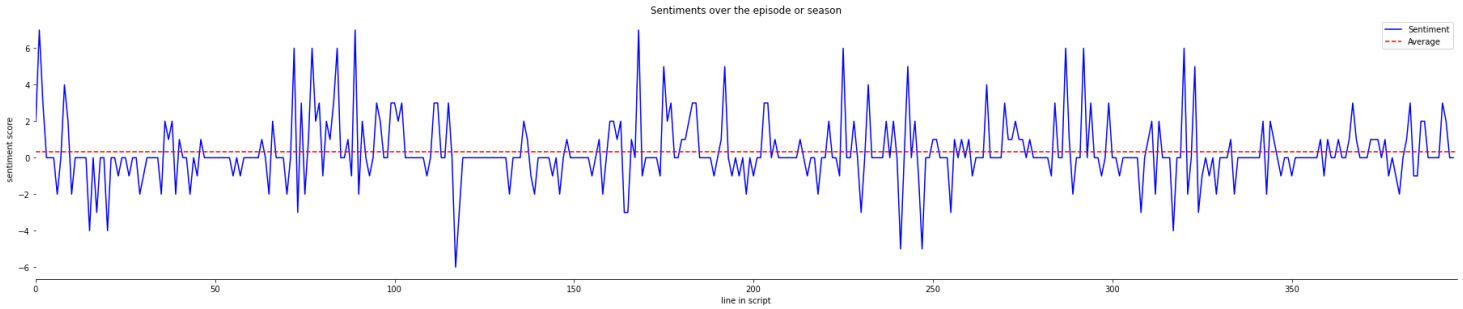
In [29]:
```
generate_overview_episode(7,15)
```

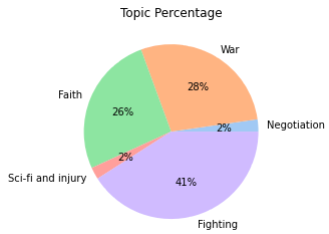
Shown is an overview over season 7 episode 15

There are 11 characters in the season.
These characters are daniel_jackson, sarah, pete, samantha_carter, jack_o_neill, teal_c, fbi, ferretty, hammond, answerphone, and osiris.

The 3 with the highest word counts are daniel_jackson, pete, and samantha_carter.

In total, there are 3493 words spoken.
The sentiment score over the episode/season has an average of 0.3207.
The graph below shows the sentiment for each line in chronological order.





In [ ]:
```
```