MPST: Movie Plot Synopses with Tags

1. Business Problem

1.1 Description

Description

Abstract Social tagging of movies reveals a wide range of heterogeneous information about movies, like the genre, plot structure, soundtracks, metadata, visual and emotional experiences. Such information can be valuable in building automatic systems to create tags for movies. Automatic tagging systems can help recommendation engines to improve the retrieval of similar movies as well as help viewers to know what to expect from a movie in advance.

Problem Statemtent Suggest the tags based on the movie plot synopsis

1.2 Source / useful links

Data Source : https://www.kaggle.com/cryptexcode/mpst-movie-plot-synopses-with-tags
Research paper : https://www.aclweb.org/anthology/L18-1274
Useful Article :https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification

1.3 Real World / Business Objectives and Constraints

1. Predict as many tags as possible with high precision and recall.
2. Incorrect tags could impact customer experience.
3. No strict latency constraints.

2.Machine Learning problem

2.1 Data

2.1.1 Data Overview

Refer:https://www.kaggle.com/cryptexcode/mpst-movie-plot-synopses-with-tags#mpst_full_data.csv All of the data is in file mpst_full_data.csv

Data Field Explaination It contains below columns Column name - Description

mdb_idIMDB -id of the movie title -Title of the move plot_synopsis -Plot Synopsis of the move tags -Tags assigned to the move split -Position of the movie in the standard data split synopsis_source -From where the plot synopsis was collected

Size -75 MB after extractiong from zip format Number of rows : 14,828

Here we will be ignoring the split info and will be spliiting the data randomly

2.1.2 Example Data point

Plot_synopsis: Note: this synopsis is for the orginal Italian release with the segments in this certain order.Boris Karloff introduces three horror tales of the macabre and the supernatural known as the \'Three Faces of Fear\'.THE TELEPHONERosy (Michele Mercier) is an attractive, high-priced Parisian call-girl who returns to her spacious, basement apartment after an evening out when she immediately gets beset by a series of strange phone calls. The caller soon identified himself as Frank, her ex-pimp who has recently escaped from prison. Rosy is terrified for it was her testimony that landed the man in jail. Looking for solace, Rosy phones her lesbian lover Mary (Lynda Alfonsi). The two women have been estranged for some time, but Rosy is certain that she is the only one who can help her. Mary agrees to come over that night. Seconds later, Frank calls again, promising that no matter who she calls for protection,

help him. Mary agrees to come over that night. Seconds later, Frank calls again, promising that no matter who she calls for protection, he will have his revenge. Unknown to Rosy, Mary is the caller impersonating Frank. Marry arrives at Rosy\'s apartment soon after, and does her best to calm Rosy\'s nerves. She gives the panic-struck woman a tranquillizer and puts her to bed.Later that night as Rosy sleeps, Mary gets up out of bed, and pens a note of confession: she was the one making the strange phone calls when she learned of Franks escape from prison. Knowing that Rosy would call on her for help, she explains that she felt it was her way of coming back into her life after their breakup. While she is busy writing, she fails to notice an intruder in the apartment. This time it is Frank, for real. He creeps up behind Mary and strangles her to death with one of Rosys nylon stockings. The sound of the struggle awaken Rosy and she gasps in fright. The murderous pimp realizes that he just killed the wrong woman, and slowly makes his way to Rosy\'s bed. However, earlier that night, Rosy had placed a butcher knife under her pillow at Mary\'s suggestion. Rosy seizes the knife and stabs Frank with it as he\'s beginning to strangle her. Rosy drops the knife and breaks down in hysteria, surrounded by the two corpses of her former lovers.THE WURDALAKIn 19th Century Russia, Vladimir D\'Urfe is a young nobleman on a long trip. During the course of his journey, he finds a beheaded corpse with a knife plunged into its heart. He withdraws the blade and takes it as a souvenir.Later that night, Vladimir stops at a small rural cottage to ask for shelter. He notices several daggers hanging up on one of the walls, and a vacant space that happens to fit the one he has discovered. Vladimir is surprised by the entrance of Giorgio (Glauco Onorato), who explains that the knife belongs to his father, who has not been seen for five days. Giorgio offers a room to the young count, and subsequently introduces him to the rest of the family: his wife (Rika Dialina), their young son Ivan, Giorgio\'s younger brother Pietro (Massimo Righi), and sister Sdenka (Susy Anderson). It subsequently transpires that they are eagerly anticipating the arrival of their father, Gorcha, as well as the reason for his absence: he has gone to do battle with the outlaw and dreaded wurdalak Ali Beg. Vladimir is confused by the term, and Sdenka explains that a wurdalak is a walking cadaver who feeds on the blood of the living, preferably close friends and family members. Giorgio and Pietro are certain that the corpse Vladimir had discovered is that of Ali Beg, but also realize that there is a strong possibility that their father has been infected by the blood curse too. They warn the count to leave, but he decides to stay and await the old mans return.At the stroke of midnight, Gorcha (Boris Karloff) returns to the cottage. His sour demeanor and unkempt appearance bode the worse, and the two brothers are torn: they realize that it is their duty to kill Gorcha before he feeds on the family, but their love for him makes it difficult to reach a decision. Later that night, both Ivan and Pietro are attacked by Gorcha who drains them of blood, and then flees the cottage. Giorgio stakes and beheads Pietro to prevent him from reviving as a wurdalak. But he is prevented from doing so to Ivan when his wife threatens to commit suicide. Reluntantly, he agrees to bury the child without taking the necessary precautions.That same night, the child rises from his grave and begs to be invited into the cottage. The mother runs to her son\'s aid, stabbing Giorgio when he attempts to stop her, only to be greeted at the front door by Gorcha. The old man bits and infects his daughter-in-law, who then does the same for her husband. Vladimir and Sdenka flee from the cottage and go on the run and hide out in the ruins of an abandoned cathedral as dawn breaks. Vladimir is optimistic that a long and happy life lies with them. But Sdenka is reluctant to relinquish her family ties. She believes that she is meant to stay with the family.Sdenka\'s fears about her family are confirmed when that evening, Gorcha and her siblings show up at the abandoned Abby. As Vladimir sleeps, Sdenka is lured into their loving arms where they bite to death. Awakened by her screams, Vladimir rushes to her aid, but the family has already taken her home, forcing the lover to follow suite. The young nobleman finds her, lying motionless on her bed. Sdenka awakens, and a distinct change is visible on her face. No longer caring, Vladimir embraces her, and she bites and infects him too.THE DROP OF WATERIn Victorian London, England, Nurse Helen Chester (Jacqueline Pierreux) is called to a large house to prepare the corpse of an elderly medium for her burial. As she dressed the body, she notices an elaborate diamond ring on its finger. Tempted by greed, Nurse Chester steals it. As she does, a glass tips over, and drops of water begin to splash on the floor. She is also assailed by a fly, no doubt attracted by the odor of the body. Unsettled but pleased by her acquisition, she finishes the job and returns home to her small East End flat.After returning home, Nurse Chester is assailed by strange events. The buzzing fly returns and continues to pester her. Then the lights in her apartment go out, and the sounds of the dripping water continues with maddening regularity. She sees the old womans corpse lying on her bed, and coming towards her. The terrified woman begs for forgiveness, but she ultimately strangles herself, imaging that the medium\'s hands are gripping her throat.The next morning, the concierge (Harriet White Medin) discovers Nurse Chester\'s body and calls the police. The investigator on the scene (Gustavo de Nardo) quickly concludes that its a simple case and that Nurse Chester "died of fright". The pathologist arrives on the scene to examine the body before it\'s taken away and he notes that the only sign of violence is a small bruise on her left finger, mostly likely caused when someone pried a ring from her finger. As the doctor makes this observation, the concierge appears distressed, as she has apparently took the ring from the dead Nurse Chester, and is further distracted by the sound of a fly swooping about in the air....Boris Karloff makes a final appearance as Gorcha riding on his horse as he concludes the three tales of fear and tells the viewers to be careful while walking home at night for ghosts and vampires have no fear. The image pulls back to actually reveal him sitting on a prop fake horse with a camera crew and various crewmen moving branches around to simulate the scene of riding through the forest from the Wurdalak segment.'

Tags:cult, horror, gothic, murder, atmospheric

2.2 Mapping the real-world problem to a Machine Learning Problem

2.2.1 Type of Machine Learning Problem

It is a multi-label classification problem **Multi-label Classification**: Multilabel classification assigns to each sample a set of target labels. This can be thought as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A question on Stackoverflow might be about any of C, Pointers, FileIO and/or memory-management at the same time or none of these. **Credit**: http://scikit-learn.org/stable/modules/multiclass.html

2.2.2 Performance metric

Micro-Averaged F1-Score (Mean F Score) : The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:
F1 = 2 *(precision* recall) / (precision + recall)
In the multi-class and multi-label case, this is the weighted average of the F1 score of each class.
'Micro f1 score':
Calculate metrics globally by counting the total true positives, false negatives and false positives. This is a better metric when we have class imbalance. ***'Macro f1 score':***
Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.
https://www.kaggle.com/wiki/MeanFScore
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
Hamming loss : ** The Hamming loss is the fraction of labels that are incorrectly predicted.
https://www.kaggle.com/wiki/HammingLoss

3. Exploratory Data Analysis

3.1 Data Loading and Cleaning

In [3]:

```
#importing required libraries
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the i
nput directory
import matplotlib.pyplot as plt
import os
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.multiclass import OneVsRestClassifier
from sklearn.linear_model import SGDClassifier
from sklearn import metrics
from sklearn.metrics import f1_score,precision_score,recall_score
from sklearn import svm
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB

import warnings
warnings.filterwarnings('ignore')

# Any results you write to the current directory are saved as output.
```

3.1.1 Using Pandas to Load the data

In [5]:

```
data = pd.read_csv('mpst_full_data.csv')
data.head()
```

Out[5]:

	imdb_id	title	plot_synopsis	tags	split	synopsis_source
0	tt0057603	I tre volti della paura	Note: this synopsis is for the orginal Italian...	cult, horror, gothic, murder, atmospheric	train	imdb
1	tt1733125	Dungeons & Dragons: The Book of Vile Darkness	Two thousand years ago, Nhagruul the Foul, a s...	violence	train	imdb
2	tt0033045	The Shop Around the Corner	Matuschek's, a gift store in Budapest, is the ...	romantic	test	imdb
3	tt0113862	Mr. Holland's Opus	Glenn Holland, not a morning person by anyone'...	inspiring, romantic, stupid, feel-good	train	imdb
4	tt0086250	Scarface	In May 1980, a Cuban man named Tony Montana (A...	cruelty, murder, dramatic, cult, violence, atm...	val	imdb

3.1.2 Counting the number of rows

In [6]:

```
print('There are {} rows in the data set'.format(data.shape[0]))
```

```
There are 14828 rows in the data set
```

3.1.3 Checking for duplicates

In [7]:

```
dup_data = data[data.duplicated(['title','plot_synopsis','tags'], keep=False)]
dup_data = dup_data.sort_values('title', axis=0, ascending=True, inplace=False, kind='quicksort',
na_position='last')
dup_data
```

Out[7]:

	imdb_id	title	plot_synopsis	tags	split	synopsis_source
717	tt0087056	A Christmas Carol	Dickens divided the book into five chapters, w...	horror	train	wikipedia
5461	tt1067106	A Christmas Carol	Dickens divided the book into five chapters, w...	horror	train	wikipedia
9281	tt0000966	A Midsummer Night's Dream	The play consists of four interconnecting plot...	romantic, fantasy	train	wikipedia
10336	tt0026714	A Midsummer Night's Dream	The play consists of four interconnecting plot...	romantic, fantasy	val	wikipedia
7559	tt0140379	A Midsummer Night's Dream	The play consists of four interconnecting plot...	romantic, fantasy	train	wikipedia
8988	tt0081595	A Tale of Two Cities	=== Book the First: Recalled to Life ===\nDick...	romantic	train	wikipedia
9991	tt0008652	A Tale of Two Cities	=== Book the First: Recalled to Life ===\nDick...	romantic	test	wikipedia
10368	tt0052270	A Tale of Two Cities	=== Book the First: Recalled to Life ===\nDick...	romantic	train	wikipedia
771	tt0001915	A Tale of Two Cities	=== Book the First: Recalled to Life ===\nDick...	romantic	train	wikipedia
11060	tt0085176	Antony and Cleopatra	Mark Antony – one of the triumvirs of the Roma...	tragedy	val	wikipedia
14340	tt0175447	Antony and Cleopatra	Mark Antony – one of the triumvirs of the Roma...	tragedy	train	wikipedia
8085	tt0043289	Antony and Cleopatra	Mark Antony – one of the triumvirs of the Roma...	tragedy	test	wikipedia
7523	tt0000634	Antony and Cleopatra	Mark Antony – one of the triumvirs of the Roma...	tragedy	val	wikipedia
13090	tt0062692	Baazi	Madan (Dev Anand) comes from a once well-to-do...	romantic, murder	train	wikipedia
10812	tt0043307	Baazi	Madan (Dev Anand) comes from a once well-to-do...	romantic, murder	test	wikipedia
10401	tt0092626	Beauty and the Beast	A widower merchant lives in a mansion with his...	romantic, fantasy	train	wikipedia
12768	tt1473343	Beauty and the Beast	A widower merchant lives in a mansion with his...	fantasy	val	wikipedia
14213	tt2771200	Beauty and the Beast	A widower merchant lives in a mansion with his...	fantasy	train	wikipedia
9600	tt0055781	Beauty and the Beast	A widower merchant lives in a mansion with his...	fantasy	test	wikipedia
9169	tt0278343	Beauty and the Beast	A widower merchant lives in a mansion with his...	fantasy	train	wikipedia
9051	tt0024866	Beauty and the Beast	A widower merchant lives in a mansion with his...	fantasy	train	wikipedia
8479	tt0074193	Beauty and the Beast	A widower merchant lives in a mansion	fantasy	test	wikipedia

	imdb_id	title	plot_synopsis	tags	split	synopsis_source
14686	tt1202150	Beauty and the Beast	A widower merchant lives in a mansion with his...	romantic, fantasy	train	wikipedia
9493	tt2380408	Clear History	In 2003, bearded, long-haired Nathan Flomm (La...	revenge	train	wikipedia
7207	tt2279864	Clear History	In 2003, bearded, long-haired Nathan Flomm (La...	revenge	train	wikipedia
12927	tt0078999	Coriolanus	The play opens in Rome shortly after the expul...	tragedy	train	wikipedia
7764	tt1372686	Coriolanus	The play opens in Rome shortly after the expul...	tragedy	train	wikipedia
6728	tt1937137	Devil's Canyon	Arizona, 1897: A female outlaw, Abby Nixon, wa...	revenge	val	wikipedia
10520	tt0045684	Devil's Canyon	Arizona, 1897: A female outlaw, Abby Nixon, wa...	revenge	train	wikipedia
10316	tt0286598	Doom	In the year 2046, a heavily populated research...	violence	train	wikipedia
...	...	...	...	...	...	...
9143	tt0061075	Ten Little Indians	Ten people travel by aerial tramway to a snowb...	murder	test	wikipedia
2624	tt0061534	The Crucible	=== Act One ===\nBetty Parris, the ten-year-ol...	allegory	train	wikipedia
10484	tt0080568	The Crucible	=== Act One ===\nBetty Parris, the ten-year-ol...	allegory	train	wikipedia
9122	tt0020815	The Dawn Patrol	During World War I, the pilots of an RFC squad...	revenge	train	wikipedia
9407	tt0030044	The Dawn Patrol	During World War I, the pilots of an RFC squad...	revenge	train	wikipedia
5958	tt0116277	The Fan	Gaurav Chandana (Shah Rukh Khan) is a Delhi-ba...	murder	train	wikipedia
9724	tt0082362	The Fan	Gaurav Chandana (Shah Rukh Khan) is a Delhi-ba...	murder	val	wikipedia
6276	tt0085692	The Hound of the Baskervilles	Dr James Mortimer asks Sherlock Holmes to inve...	murder, haunting	train	wikipedia
7824	tt0068719	The Hound of the Baskervilles	Dr James Mortimer asks Sherlock Holmes to inve...	murder, haunting	test	wikipedia
12686	tt0186264	The Karate Kid	High school senior Daniel LaRusso and his moth...	violence	test	wikipedia
11983	tt0137311	The Karate Kid	High school senior Daniel LaRusso and his moth...	violence	train	wikipedia
3524	tt2140553	The Last of Us	In September 2013, an outbreak of a mutant Cor...	violence	train	wikipedia
12819	tt3581920	The Last of Us	In September 2013, an outbreak of a mutant Cor...	violence	train	wikipedia
9696	tt0114345	The Scarlet Letter	In June 1642, in the Puritan town of Boston, a...	historical fiction	train	wikipedia
9218	tt0025747	The Scarlet Letter	In June 1642, in the Puritan town of Boston, a...	historical fiction	test	wikipedia
12439	tt0182408	The Scarlet Pimpernel	In 1792, at the bloody height of the French Re...	action, historical fiction	val	wikipedia
9953	tt0025748	The Scarlet Pimpernel	In 1792, at the bloody height of the French Re...	action, historical fiction	val	wikipedia
9999	tt0084637	The Scarlet Pimpernel	In 1792, at the bloody height of the French Re...	action, historical fiction	val	wikipedia
12060	tt0029543	The Shadow	In Tibet, following the First World War, an Am...	murder	val	wikipedia
9654	tt0028243	The Shadow	In Tibet, following the First World War, an Am...	murder	val	wikipedia
11216	tt0022428	The Squaw Man	James Wynnegate (Dustin Farnum) and his cousin...	murder	train	wikipedia
14617	tt0004635	The Squaw Man	James Wynnegate (Dustin Farnum) and his cousin...	murder	train	wikipedia
14239	tt0032028	The Three Musketeers	In Venice, the musketeers Athos, Porthos, and ...	good versus evil, action, historical fiction	train	wikipedia
7723	tt0012752	The Three Musketeers	In Venice, the musketeers Athos, Porthos,	good versus evil, action,	val	wikipedia

	imdb_id	title	plot_synopsis	tags	split	synopsis_source
			and ...	historical fiction		
7119	tt0209440	The Turn of the Screw	An unnamed narrator listens to Douglas, a frie...	insanity, haunting	train	wikipedia
14074	tt0105659	The Turn of the Screw	An unnamed narrator listens to Douglas, a frie...	insanity, haunting	train	wikipedia
12713	tt0758796	The Violent Kind	Cody, Q, and Elroy are second-generation biker...	violence	train	wikipedia
14448	tt1472195	The Violent Kind	Cody, Q, and Elroy are second-generation biker...	violence	train	wikipedia
8924	tt0043086	Under the Gun	At a Miami nightclub, gangster Bert Galvin off...	murder	train	wikipedia
6055	tt5275886	Under the Gun	At a Miami nightclub, gangster Bert Galvin off...	murder	train	wikipedia

134 rows × 6 columns

We can see many duplicates in the data lets keep one point and remove rest

In [8]:

```
#dropping the duplicates
clean_data=data.drop_duplicates(subset={'title','plot_synopsis','tags'}, keep='first',
inplace=False)
print('Number of data points before removing duplicates',data.shape[0])
print('Number of data points after removing duplicates',clean_data.shape[0],' and duplicates are a
bout {}% from original data'
      .format((1-(clean_data.shape[0]/data.shape[0]))*100))
```

```
Number of data points before removing duplicates 14828
Number of data points after removing duplicates 14752  and duplicates are about 0.512543835985968%
from original data
```

3.2 Analysis of Tags

3.2.1 Total number of unique tags

In [9]:

```
total_tags = []
for sent in clean_data['tags'].values:
    for tag in sent.split(','):
            total_tags.append(tag)
tags = pd.DataFrame(total_tags,columns=['tags'])
freq = tags['tags'].value_counts().to_dict()
tag_df = pd.DataFrame(list(freq.items()),columns=['tags','counts'])
print('Total number of unique tags',tag_df.shape[0])
```

```
Total number of unique tags 142
```

3.2.2 Avg number of tags per question

In [10]:

```
avg_tag = np.array(total_tags).shape[0]/clean_data.shape[0]
print('Avg tag per question',avg_tag)
```

```
Avg tag per question 2.9900352494577005
```

3.2.3 MAx and Min number of tags per question

In [11]:

```
length= []
for i,x in clean_data['tags'].items():
    j = x.split(',')
    length.append(len(j))
clean_data['tags_freq'] = length

print('Maximim number of tags per question',clean_data['tags_freq'].max())
print('Minimum number of tags per question',clean_data['tags_freq'].min())
```

```
Maximim number of tags per question 25
Minimum number of tags per question 1
```

3.2.4 Number of times a tag appeared

In [12]:

```
tag_df.head()
```

Out[12]:

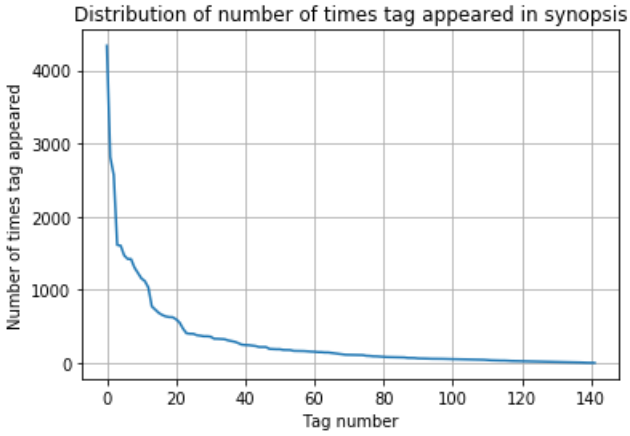
	tags	counts
0	murder	4338
1	violence	2814
2	flashback	2571
3	cult	1614
4	violence	1606

In [13]:

```
tag_df_sorted = tag_df.sort_values(['counts'], ascending=False)
tag_counts = tag_df_sorted['counts'].values
```

In [14]:

```
plt.plot(tag_counts)
plt.title("Distribution of number of times tag appeared in synopsis")
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
```

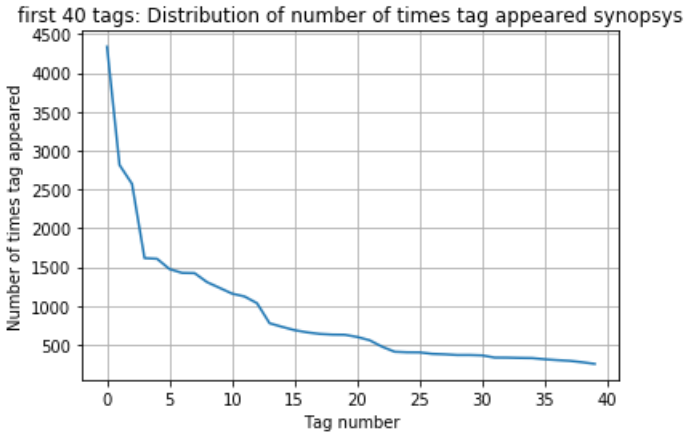


We can see tag number less than 40 appeared many times . Lets see distribution for first 40

In [15]:

```
plt.plot(tag_counts[0:40])
plt.title('first 40 tags: Distribution of number of times tag appeared synopsys')
plt.grid()
```

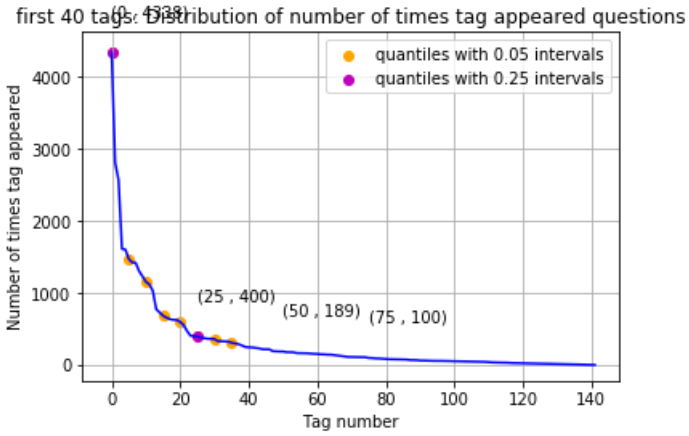
```
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.show()
print(len(tag_counts[0:40:5]), tag_counts[0:40:5])
```




```
8 [4338 1473 1157  686  599  400  361  311]
```

In [16]:

```
plt.plot(tag_counts, c='b')
plt.scatter(x=list(range(0,40,5)), y=tag_counts[0:40:5], c='orange', label="quantiles with 0.05 int
ervals")
# quantiles with 0.25 difference
plt.scatter(x=list(range(0,40,25)), y=tag_counts[0:40:25], c='m', label = "quantiles with 0.25 inte
rvals")

for x,y in zip(list(range(0,100,25)), tag_counts[0:100:25]):
    plt.annotate(s="({} , {})".format(x,y), xy=(x,y), xytext=(x-0.05, y+500))

plt.title('first 40 tags: Distribution of number of times tag appeared questions')
plt.grid()
plt.xlabel("Tag number")
plt.ylabel("Number of times tag appeared")
plt.legend()
plt.show()
print(len(tag_counts[0:40:5]), tag_counts[0:40:5])
```




```
8 [4338 1473 1157  686  599  400  361  311]
```

In [17]:

```
lst_tags_gt_10k = tag_df[tag_df.counts>1000].tags
#Print the length of the list
print ('{} Tags are used more than 1000 times'.format(len(lst_tags_gt_10k)))
# Store tags greater than 100K in one list
lst_tags_gt_100k = tag_df[tag_df.counts>500].tags
#Print the length of the list.
```

```
print ('{} Tags are used more than 500 times'.format(len(lst_tags_gt_100k)))
```

```
13 Tags are used more than 1000 times
22 Tags are used more than 500 times
```

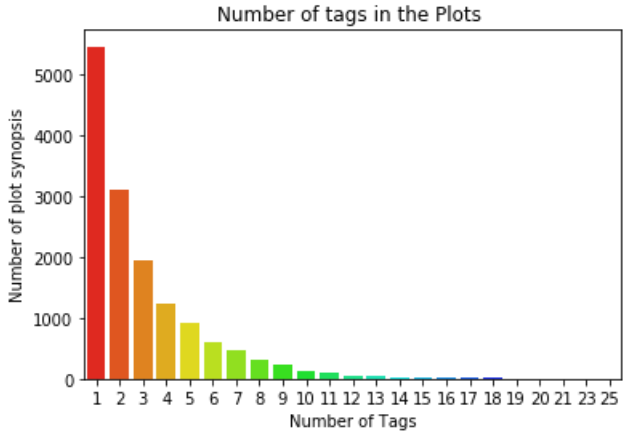
Observations:

1. There are total 13 tags which are used more than 1000 times.
2. 22 tags are used more than 500 times.
3. Most frequent tag (i.e. Murderer) is used 4338 times.
4. Since some tags occur much more frequenctly than others, Micro-averaged F1-score is the appropriate metric for this probelm

3.2.4 Tags Per Question

In [18]:

```
import seaborn as sns
sns.countplot(clean_data['tags_freq'], palette='gist_rainbow')
plt.title("Number of tags in the Plots ")
plt.xlabel("Number of Tags")
plt.ylabel("Number of plot synopsis")
plt.show()
```

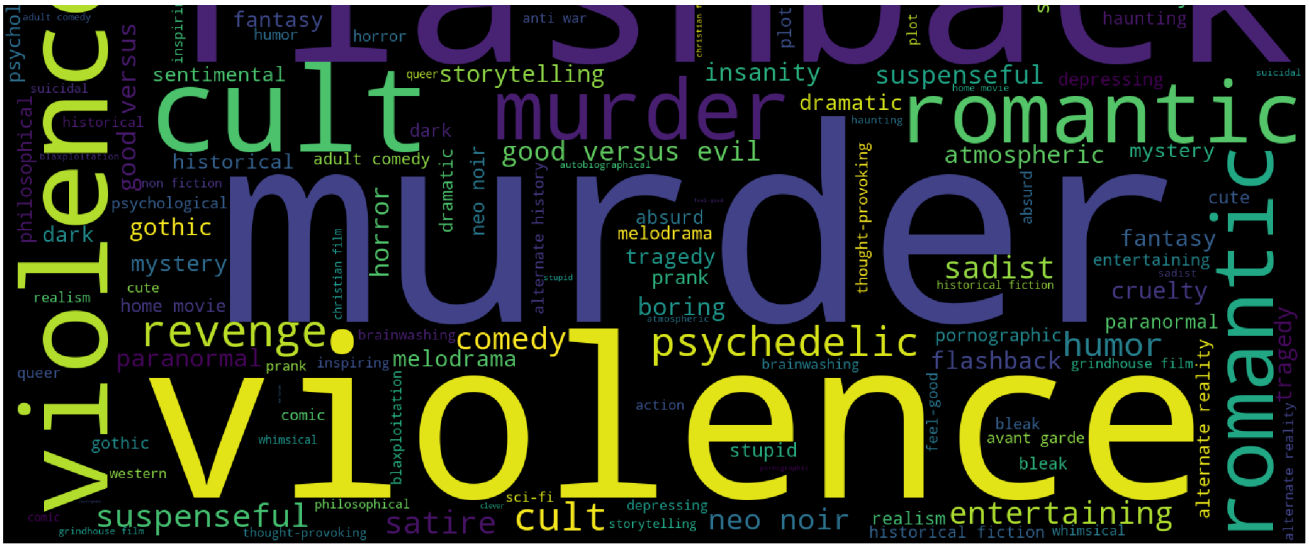


we can see many plots contains only single tag

3.2.5 Most Frequent Tags

In [19]:

```
# Ploting word cloud
from datetime import datetime
from wordcloud import WordCloud
start = datetime.now()
# Lets first convert the 'result' dictionary to 'list of tuples'
tup = dict(tag_df.items())
#Initializing WordCloud using frequencies of tags.
wordcloud = WordCloud( background_color='black',width=1600,height=800,).generate_from_frequencies(
freq)
fig = plt.figure(figsize=(30,20))
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)

plt.show()
print("Time taken to run this cell :", datetime.now() - start)
```

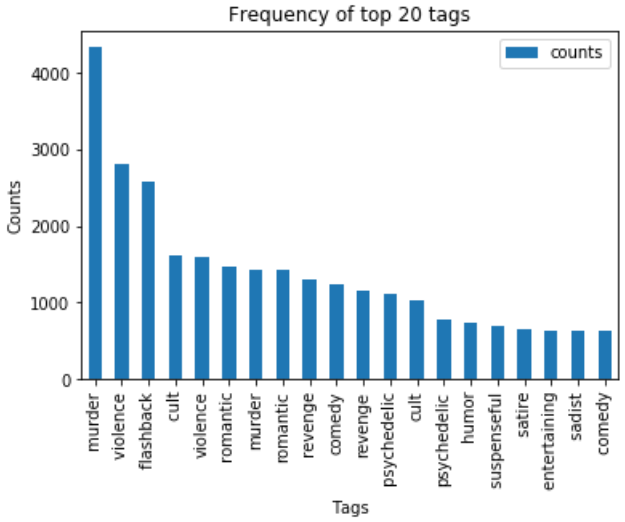



```
Time taken to run this cell : 0:00:08.753596
```

3.2.6 The top 20 tags

In [21]:

```
i=np.arange(20)
tag_df_sorted.head(20).plot(kind='bar')
plt.title('Frequency of top 20 tags')
plt.xticks(i, tag_df_sorted['tags'])
plt.xlabel('Tags')
plt.ylabel('Counts')
plt.show()
```



Observations:

1. Majority of the most frequent tags are Murder,violence,Flashback.
2. Murder is the top most frequent programming language.

3.3 Cleaning and preprocessing of Plot Synopsis

Now that we have finished deduplication our data requires some preprocessing before we go on further with analysis and making the prediction model.

Hence in the Preprocessing phase we do the following in the order below:-

1. Begin by removing the html tags

1. Begin by removing the html tags
2. Remove any punctuations or limited set of special characters like , or . or # etc.
3. Check if the word is made up of english letters and is not alpha-numeric
4. Convert the word to lowercase
5. Remove Stopwords

In [22]:

```
# printing some random reviews
sent_0 = data['plot_synopsis'].values[0]
print(sent_0)
print("="*50)
sent_1000 = data['plot_synopsis'].values[1000]
print(sent_1000)
print("="*50)
sent_1500 = data['plot_synopsis'].values[1500]
print(sent_1500)
print("="*50)
sent_4900 = data['plot_synopsis'].values[4900]
print(sent_4900)
```

```
Note: this synopsis is for the orginal Italian release with the segments in this certain
order.Boris Karloff introduces three horror tales of the macabre and the supernatural known as the
'Three Faces of Fear'.THE TELEPHONERosy (Michele Mercier) is an attractive, high-priced Parisian c
all-girl who returns to her spacious, basement apartment after an evening out when she immediately
gets beset by a series of strange phone calls. The caller soon identified himself as Frank, her ex
-pimp who has recently escaped from prison. Rosy is terrified for it was her testimony that landed
the man in jail. Looking for solace, Rosy phones her lesbian lover Mary (Lynda Alfonsi). The two w
omen have been estranged for some time, but Rosy is certain that she is the only one who can help
her. Mary agrees to come over that night. Seconds later, Frank calls again, promising that no matt
er who she calls for protection, he will have his revenge. Unknown to Rosy, Mary is the caller imp
ersonating Frank. Marry arrives at Rosy's apartment soon after, and does her best to calm Rosy's n
erves. She gives the panic-struck woman a tranquillizer and puts her to bed.Later that night as Ro
sy sleeps, Mary gets up out of bed, and pens a note of confession: she was the one making the stra
nge phone calls when she learned of Franks escape from prison. Knowing that Rosy would call on her
for help, she explains that she felt it was her way of coming back into her life after their break
up. While she is busy writing, she fails to notice an intruder in the apartment. This time it is F
rank, for real. He creeps up behind Mary and strangles her to death with one of Rosys nylon
stockings. The sound of the struggle awaken Rosy and she gasps in fright. The murderous pimp reali
zes that he just killed the wrong woman, and slowly makes his way to Rosy's bed. However, earlier
that night, Rosy had placed a butcher knife under her pillow at Mary's suggestion. Rosy seizes the
knife and stabs Frank with it as he's beginning to strangle her. Rosy drops the knife and breaks d
own in hysteria, surrounded by the two corpses of her former lovers.THE WURDALAKIn 19th Century Ru
ssia, Vladimir D'Urfe is a young nobleman on a long trip. During the course of his journey, he fin
ds a beheaded corpse with a knife plunged into its heart. He withdraws the blade and takes it as a
souvenir.Later that night, Vladimir stops at a small rural cottage to ask for shelter. He notices
several daggers hanging up on one of the walls, and a vacant space that happens to fit the one he
has discovered. Vladimir is surprised by the entrance of Giorgio (Glauco Onorato), who explains th
at the knife belongs to his father, who has not been seen for five days. Giorgio offers a room to
the young count, and subsequently introduces him to the rest of the family: his wife (Rika
Dialina), their young son Ivan, Giorgio's younger brother Pietro (Massimo Righi), and sister Sdenk
a (Susy Anderson). It subsequently transpires that they are eagerly anticipating the arrival of
their father, Gorcha, as well as the reason for his absence: he has gone to do battle with the out
law and dreaded wurdalak Ali Beg. Vladimir is confused by the term, and Sdenka explains that a wur
dalak is a walking cadaver who feeds on the blood of the living, preferably close friends and
family members. Giorgio and Pietro are certain that the corpse Vladimir had discovered is that of
Ali Beg, but also realize that there is a strong possibility that their father has been infected b
y the blood curse too. They warn the count to leave, but he decides to stay and await the old mans
return.At the stroke of midnight, Gorcha (Boris Karloff) returns to the cottage. His sour demeanor
and unkempt appearance bode the worse, and the two brothers are torn: they realize that it is thei
r duty to kill Gorcha before he feeds on the family, but their love for him makes it difficult to
reach a decision. Later that night, both Ivan and Pietro are attacked by Gorcha who drains them of
blood, and then flees the cottage. Giorgio stakes and beheads Pietro to prevent him from reviving
as a wurdalak. But he is prevented from doing so to Ivan when his wife threatens to commit
suicide. Reluntantly, he agrees to bury the child without taking the necessary precautions.That sa
me night, the child rises from his grave and begs to be invited into the cottage. The mother runs
to her son's aid, stabbing Giorgio when he attempts to stop her, only to be greeted at the front d
oor by Gorcha. The old man bits and infects his daughter-in-law, who then does the same for her hu
sband. Vladimir and Sdenka flee from the cottage and go on the run and hide out in the ruins of an
abandoned cathedral as dawn breaks. Vladimir is optimistic that a long and happy life lies with th
em. But Sdenka is reluctant to relinquish her family ties. She believes that she is meant to stay
with the family.Sdenka's fears about her family are confirmed when that evening, Gorcha and her si
blings show up at the abandoned Abby. As Vladimir sleeps, Sdenka is lured into their loving arms w
here they bite to death. Awakened by her screams, Vladimir rushes to her aid, but the family has a
lready taken her home, forcing the lover to follow suite. The young nobleman finds her, lying
motionless on her bed. Sdenka awakens, and a distinct change is visible on her face. No longer car
```

motionless on her bed. Sdenka awakens, and a distinct change is visible on her face. No longer cari
ing, Vladimir embraces her, and she bites and infects him too.THE DROP OF WATERIn Victorian London
, England, Nurse Helen Chester (Jacqueline Pierreux) is called to a large house to prepare the
corpse of an elderly medium for her burial. As she dressed the body, she notices an elaborate diam
ond ring on its finger. Tempted by greed, Nurse Chester steals it. As she does, a glass tips over,
and drops of water begin to splash on the floor. She is also assailed by a fly, no doubt attracted
by the odor of the body. Unsettled but pleased by her acquisition, she finishes the job and return
s home to her small East End flat.After returning home, Nurse Chester is assailed by strange
events. The buzzing fly returns and continues to pester her. Then the lights in her apartment go o
ut, and the sounds of the dripping water continues with maddening regularity. She sees the old wom
ans corpse lying on her bed, and coming towards her. The terrified woman begs for forgiveness, but
she ultimately strangles herself, imaging that the medium's hands are gripping her throat.The next
morning, the concierge (Harriet White Medin) discovers Nurse Chester's body and calls the police.
The investigator on the scene (Gustavo de Nardo) quickly concludes that its a simple case and that
Nurse Chester "died of fright". The pathologist arrives on the scene to examine the body before it
's taken away and he notes that the only sign of violence is a small bruise on her left finger, mo
stly likely caused when someone pried a ring from her finger. As the doctor makes this
observation, the concierge appears distressed, as she has apparently took the ring from the dead N
urse Chester, and is further distracted by the sound of a fly swooping about in the air....Boris K
arloff makes a final appearance as Gorcha riding on his horse as he concludes the three tales of
fear and tells the viewers to be careful while walking home at night for ghosts and vampires have
no fear. The image pulls back to actually reveal him sitting on a prop fake horse with a camera cr
ew and various crewmen moving branches around to simulate the scene of riding through the forest f
rom the Wurdalak segment.
==========
The movie opens with a montage of crazed customers storming into a store for their Christmas
shopping, trampling over the store's employees before maniacally grabbing every hot item they can
get their hands on, all to the tune of "It's Beginning to Look a Lot Like Christmas". In the mall,
we see two parents, Tom and Sarah Engel (Adam Scott and Toni Collette), run to their son Max (Emja
y Anthony) as he is fighting with another child after this child spoke ill of Santa Claus to the
other children. Max's sister Beth (Stefania LaVie Owen) gleefully films everything on her
phone.The family returns home where Max's German grandmother Omi (Krista Stadler) is baking. Sarah
continues working in the kitchen in anticipation of her sister and her family arriving. She is anx
ious, as is everyone else, but Sarah tries to keep the Christmas spirit alive in the house. She ha
ngs up a picture of her family posing with Santa, but it's ruined by everyone looking uninterested
and Santa checking out Beth's butt.Sarah's sister Linda (Allison Tolman) arrives with her husband
Howard (David Koechner) and their kids - Howie Jr (Maverick Flack), Stevie (Lolo Owen), Jordan (Qu
eenie Samuel), and Baby Chrissy, along with the family dog Rosie. Also joining them, to Sarah's ex
treme displeasure, is their Aunt Dorothy (Conchata Ferrell), whom Sarah considers to be a
nightmare. The family gathers for dinner, where Howard takes the time to boast about his use of gu
ns, despite Linda telling him not to bring that up. Dorothy insults Sarah's cooking, so Sarah goes
to get the dessert in the kitchen, where Dorothy follows her to make more complaints until Sarah s
huts her up. Stevie and Jordan mock Max for still writing a letter to Santa. They swiped it off of
him and start reading off Max's wishes. He wants his parents to be in love again, for him and Beth
to spend time like they used to, and for things to go better between Linda and Howard. Stevie gets
angry when she reads that Max wrote that Howard wishes she and Jordan were boys. Max loses it and
fights Stevie to get the letter back. He yells at the whole family and says he wishes Christmas wa
s like it used to be, but now he hates the holiday and his dysfunctional family before running bac
k upstairs.Tom goes upstairs to comfort his son and remind him that this is the time of year where
they have to deal with family members like this. After Tom leaves, Max puts his Santa letter into
an envelope, but he is too angry to go through with it and he tears up the letter. He throws the p
ieces out the window, which are then blown up high into the sky. A dark cloud then forms over the
whole neighborhood, followed by a strong gust of wind that takes out the power.The next morning, t
he whole family is struggling without heat and electricity as a snowstorms blows through the area.
Max looks out the window and notices a creepy snowman having been built in front of the house. Som
eone knocks on the door and Linda answers it to find a delivery guy bringing some boxes. Next to h
im is a sack of gifts, but he claims it didn't come from him. Beth suggests that she walk a few bl
ocks to her boyfriend Derek's house to see if he has power (and also so she won't be stuck with th
e family). Although Sarah is hesitant, she and Tom allow Beth to go out for an hour.Beth walks thr
ough the storm and sees a large horned creature perched on top of a roof. The sky becomes darker,
and the creature appears to follow Beth. She runs away as the creature pounces across rooftops. Be
th finds the delivery guy literally frozen in fear. She hides underneath his truck as the creature
lands on the ground, where Beth sees its hooves. To her side, she sees a jack-in-the-box playing a
tune. It opens and a smaller creature emerges slowly before attacking Beth.As it gets darker, Tom
and Sarah become concerned when Beth doesn't return. Howard agrees to go out with Tom in his Humme
r to find her, and Howard gives Tom a gun to carry just in case. They go to Derek's house and find
it unoccupied and wrecked. The two see large hoofprints in the snow. They go back outside and Tom
hears Beth screaming. He runs to find her, only for something burrowing underneath the snow to gra
b Howard. It nearly drags him beneath the snow until Tom shoots at the creature and sends it fleei
ng. They find the Hummer has been destroyed and they run home. The ladies tend to Howard's leg, wh
ich appears to have a bite mark on it. Omi is seen looking nervous by the fireplace. Tom says they
need to board up the house and look for Beth in the morning.The adults decide to take turns keepin
g watch for anything suspicious. Howard volunteers to stay awake while Tom rests with his family.
Howard also apologizes to Tom for thinking he was always a "spineless dick". Eventually, everyone
falls asleep and the fire goes out. A hook descends from the chimney with a gingerbread man cookie
tied to it. This awakens Howie Jr. He goes over to the cookie and takes a bite out of its head, on
ly for the gingerbread man to come to life and frighten Howie. The hook wraps around Howie and sta
rts pulling him up the tree. Sarah wakes up and runs to grab the boy, with the rest of the adults

its pulling him up the tree. Sarah wakes up and runs to grab the boy, with the rest of the adults grabbing her for help. Sarah accidentally kicks a hot log toward the Christmas tree, igniting the presents and then the whole tree. Max runs to get the fire extinguisher and puts out the fire, but Sarah gets freaked out by the cookie, causing her to let go of Howie, and he is taken up the chimney.Now convinced that something unnatural is occurring, Omi speaks English for the first time to tell the family that she knows who is doing this, and that this is all their fault. She says she experienced the same thing as a child. Through an animated sequence, we see Omi as a little girl in her old poverty-stricken village. She was holding a toy Santa and a loaf of bread, but the other villagers snatched the bread from her and fought each other for it. When Omi returned home, her parents didn't help matters through their own bad behavior, and her mother ripped the Santa doll. Omi lost her holiday spirit and threw the doll in the fireplace after wishing that her parents would go away. This wish would summon an ancient spirit that arrives to punish anyone that disrespected the Christmas spirit. the spirit's name is Krampus, and he brings multiple helpers to terrorize the misbehaved. Omi watched as Krampus and his helpers dragged her parents to their underworld, with Krampus leaving her with nothing but a bauble with his name on it as a reminder of what she's done. Omi shows the family the bauble, and to this day, she has regretted making that wish. Howard remains unconvinced that this is the work of some supernatural entity, and he grabs his shotgun to go out and find Howie. As he opens the door, he discovers a snowman with a close resemblance to Howie on the front porch. Several demonic creatures lurk behind the snowmen, and Sarah pulls Howard back inside the house.Tom devises a plan to get to the snowplow with the family and head to the mall for shelter. Meanwhile, Stevie and Jordan think they hear what sounds like Beth's voice coming from upstairs. They walk up to investigate, but then the adults hear the girls screaming. Tom, Sarah, and Linda run up to the attic and notice that the kids' presents had something burst out of them. They then come across an enormous, worm-like Jack-in-the-Box in the process of swallowing Jordan whole. Tom shoots it, causing it to flee towards the vents, where it later escapes. The adults then get attacked by demonic toys. Sarah faces her mother's angel ornament that nearly hangs her with Christmas lights, Linda is jumped by an evil teddy bear, and Tom is attacked by a robot toy.
Downstairs, Howard gets shot with a nail gun by three gingerbread men. Linda sees Stevie in the next room, driving her to grab an icicle and stab the teddy in the eye. She grabs a hatchet and swings at the angel and robot before running to Stevie. Sarah gets Howard's gun and shoots the robot to pieces. Howard shoots a lantern that burns the cookies, only for one to survive and pounce toward him, but Rosie eats it.The family goes downstairs to the living room where Max sends Rosie into the ventilation shaft to combat the clown, but the dog is consume by it off-screen. The Jack-in-the-Box clown suddenly crashes through the ceiling into the living room. Before Howard can kill it, the teddy and cherub leap down from the hole in the ceiling to attack once more. Dorothy takes the shotgun and kills the teddy and angel, but before she can finish off the clown, a group of dark elves break into the house. They take Dorothy and Chrissy, while Howard is lost when he grabs onto the clown's tail as it is sucked out through the hole in the wall. A loud thumping is heard on the roof, signifying the arrival of Krampus and causing the elves to flee. Omi stays behind while Tom, Sarah, Linda, Max, and Stevie escape. Omi then comes face-to-face with Krampus, who looks like a demonic Santa Claus with a long tongue. He opens his sack in front of Omi, and a gruesome-looking nutcracker soldier grabs her and pulls her inside.The family heads toward the snowplow, only for the creature in the snow to attack again. It first takes Tom before also getting Linda and Sarah. Max and Stevie get into the snowplow. Max tries to start it, but the elves attack and take Stevie. Max runs after her and he is faced by Krampus. He hands Max his torn-up letter that is wrapped around a bauble similar to what he gave Omi. This leads Max to realize that it's his fault that Krampus arrived. He follows the demons to a spot where they are preparing to take Stevie. Max calls to Krampus and throws the bauble back toward him. It sinks in the snow and opens up a massive hole in the earth that leads to the underworld. The elves hold Stevie over it until Max tells Krampus that he knows it's his fault that Krampus is there, and he offers to take Stevie's place. Krampus wipes Max's tear with his claw, but he and the elves start laughing as they throw Stevie into the hole. Krampus grabs Max and holds him over the hole. Max says he only wishes Christmas could be like it used to be. Krampus drops a screaming Max into the hole.Max then wakes up and finds himself in his room, and it's now Christmas Day. He goes downstairs and sees the whole family together, looking more happy and peaceful than they did a few days ago. They start to open presents, and Max hugs his parents, thinking the whole ordeal was just a bad dream. He then opens his gift to find the Krampus bauble, and suddenly everyone has an ominous look on their faces as their memories of the horrific events slowly come back to them. The camera pans out revealing that either the family is being watched through a snow globe by Krampus, along with hundreds of others in his collection, or are now held prisoner for all eternity by the evil Krampus.The evil toys then appear for one last jump scare.

==========

Season 1Criminals, including the Russian mafia, Yakuza, and Chinese, have taken advantage of Hell's Kitchen's circumstances since "the incident". Blinded as a boy in an accident that gave him heightened senses, Matt Murdock begins fighting this rising criminal element by night as a costumed vigilante while opening a law firm with his friend, Foggy Nelson. Their first client is Karen Page, a secretary for construction company Union Allied, who has been framed for the murder of her co-worker, Daniel Fisher, after accidentally uncovering a pension embezzlement scheme.
Murdock prevents Karen from being prosecuted and protects her from an assassin, before exposing the scandal through Ben Urich at The New York Bulletin. Grateful for their help, Page offers to work for Murdock and Nelson. James Wesley covers up the involvement of his employer in the scandal, and orders Anatoly and Vladamir Ranskahov, the Russians' leaders, to deal with Murdock ("the man in black"): they kidnap a young boy to lure him into a trap.When Murdock was a boy, his father was murdered for winning a match he was told to throw. Now, after failing to rescue the kidnapped boy, a severely injured Murdock is found in a dumpster by nurse Claire Temple. Nelson meanwhile attempts to comfort Page following her recent traumatic experiences. Temple takes Murdock to her apartment, tends to his wounds, and removes his mask, discovering his blindness. He refuses to reveal his name, but does reveal his heightened senses when they alert him to a Russian who is searching the apa

e, but does reveal his heightened senses when they alert him to a Russian who is searching the apa
rtment building, giving Temple time to hide Murdock and convince the man that she knows nothing. M
urdock realizes that the man did not believe her, and overpowers him, taking him to the roof. Murd
ock and Temple torture him into revealing the boy's location, before Murdock pushes him off the ro
of and into the same dumpster. He barely survives. Murdock enters the building where they are keep
ing the boy, defeats the guards, and rescues him.Wesley, having become aware of Nelson and Murdock
because of their involvement with Page during the Union Allied scandal, hires them to defend John
Healy, an assassin. Though Nelson wishes not to get involved with an obvious criminal element, Mur
dock wishes to use the case to discover who Wesley's employer is, and so accepts Wesley's offer, w
hich includes a substantial sum to ensure their silence. Page receives a similar offer from Union
Allied, who do not want her to talk to anyone else about the scandal, and threaten to sue her for
leaking company secrets to the press if she does not agree. Despite this, Page goes to Urich, whos
e editor Ellison is forcing him to write superfluous stories rather than the major crime breaks of
his youth, and offers to tell him more about the Union Allied scandal. After successfully
defending Healy, Murdock confronts him in costume and forces him to reveal Wesley's employer, Wils
on Fisk. Healy then commits suicide rather than face the consequences of this.Murdock is unable to
find any record of Fisk, and so continues to interrogate criminals, searching for answers. Wesley
informs the Ranskahovs of an offer Fisk has made to help with their operations, given their recent
failures. Angered at this apparent slight, they attempt to stop the man in black once and for all
by visiting in hospital the Russian whom Murdock threw from the roof. After the Russian tells them
of Temple, they send men to kidnap her, but she manages to call Murdock in time to alert him of he
r kidnapping. The Russians attempt to torture Murdock's name out of Temple, but Murdock arrives an
d defeats the gangsters. Seeing the aftermath of this, the Ranskahovs decide to agree to Fisk's of
fer, with Anatoly going to tell Fisk personally, by barging into the restaurant where Fisk is
having dinner with an art gallery curator named Vanessa Marianna. Fisk takes a confused Marianna h
ome, and angered at this intrusion and embarrassment, Fisk beheads Anatoly and orders Wesley to se
nd the body to Vladimir.Fisk explains the situation to his allies, including Chinese leader Madame
Gao, of whom he asks a special favor. Elena Cardenas, a renter of powerful businessman Armand Tull
ey, comes to Nelson and Murdock after Tulley, who wants to convert her apartment building, sends m
en to wreck her home. Nelson goes to Tulley's lawyers at Landman and Zack (where he and Murdock on
ce interned), represented by Marci Stahl, Nelson's ex-girlfriend. She explains that Cardenas and h
er neighbors can either take a large settlement or be evicted. While looking for complaints
against Tulley at the police station, Murdock realizes Detectives Carl Hoffman and Blake are corru
pt when he hears them kill a Russian. Murdock later incapacitates Blake and uses his cell phone to
find Vladimir. Fisk pays Turk Barrett to reveal to Vladimir that Fisk killed Anatoly, and as they
prepare for a war against Fisk, their forces are destroyed in a suicide attack by Gao's workers. V
ladimir survives, but Murdock finds him, as they are surrounded by police.Murdock takes out the po
lice when they try to kill Vladimir on Fisk's orders, and he takes Vladimir to an abandoned wareho
use, hoping for answers about Fisk, while Nelson and Cardenas are injured in the bombings. With Te
mple's help, Murdock cauterizes Vladimir's wounds, alerting a non-corrupt police officer to their
presence. The officer calls in Murdock and Vladimir's location, and the warehouse is soon
surrounded. Blake and Hoffman take control of the situation, and await Fisk's orders. Fisk talks t
o Murdock via police radio, telling him of his admiration for what Murdock is trying to do, even t
hough it clashes with Fisk's own plans to save the city. Fisk then frames the vigilante by having
a police sniper fire on other officers from the roof of the warehouse, including Blake, as Urich a
nd the media look on. Vladimir, in return for Murdock avenging Anatoly's death, gives him informat
ion on Leland Owlsley the accountant for all of Fisk's operations before giving his life so that M
urdock can escape.Murdock tracks down Owlsley, but is distracted by the arrival of an elderly man:
Murdock's mentor, Stick, who taught him to master his abilities as a child, but abandoned him when
Murdock developed an attachment to him. Now, he enlists Murdock's help in destroying Black Sky, a
weapon that the Japanese, led by Nobu, are bringing to New York. Stick agrees to refrain from kill
ing, but breaks his promise when he kills Black Sky, who is actually a young boy. After fighting i
n Murdock's apartment, Murdock defeats Stick, who agrees to leave the city. Urich, having agreed t
o help Page expose the further scandal and corruption surrounding Hell's Kitchen and Union Allied,
explains that they need proof before they can publish anything. While searching for a connection b
etween Tulley's men and Union Allied, Page is confronted by them. Nelson helps her escape them, so
she and Urich explain their investigation to him. Stick later converses with a heavily scarred man
about Murdock's role in events to come.Page and Nelson bring Murdock in on their plan, and he
agrees as long as they stop putting themselves in harm's way, and use the legal system rather than
underhanded tactics. Owlsley and Nobu, angry at being confronted by the vigilante and losing Black
Sky, respectively, express their displeasure with Fisk, while he is also dealing with Blake, who w
akes up in hospital. Fisk convinces Hoffman to kill Blake before he can speak out against Fisk, bu
t Hoffman is incapacitated by Murdock, who gets information on Fisk from Blake before he dies. Gao
visits Fisk, warning him that he will have to get everything under control if he does not want
her, Nobu, and Owlsley to side-step him. An angry Fisk is later consoled by Marianna, and he tells
her of how, as a child, he murdered his father when he was beating Fisk's mother. She convinces hi
m to stand up and go public with his intentions to save the city. This negates all of Murdock's in
formation from Blake, which Urich was going to publish.In the wake of Fisk's public revelation, Mu
rdock, Page, Nelson, and Urich begin searching for something from Fisk's past that they can use ag
ainst him. Murdock visits Marianna at her art gallery, hoping to gain insight into Fisk by speakin
g with her, and encounters Fisk himself. Nobu demands a promised city block from Fisk, who agrees
on the condition Nobu provide him with a "specialist" in return to deal with the vigilante. The bl
ock, which Fisk has bought from Tulley, is where Cardenas lives, and she is one of the few tenants
who stands in the way of Fisk gifting it to Nobu. When Cardenas is killed by a junkie, Murdock rea
lizes that Fisk is behind it, and tracks him to an abandoned warehouse, where he is confronted by
the specialist: Nobu. Murdock eventually defeats Nobu, accidentally causing him to burn alive, but
sustains serious injuries. Fisk then confronts Murdock, beating him nearly to death. Murdock barel
y escapes, only to collapse at his home in front of Nelson. After Temple tends to Murdock's wounds

y escapes, only to collapse at his home in front of Nelson.After Temple tends to Murdock's wounds, Nelson confronts him about his "blindness" and vigilante activities. Murdock explains that after e xperiencing the twisted morals of Landman and Zack, Murdock took it upon himself to confront a pae dophilic and incestuous rapist with no legal evidence against him, but who Murdock had discovered through his heightened senses. Since then, he had been doing everything he could to make the city a better place whenever the law could not. Nelson, unable to look past Murdock's lies, leaves thei r firm. Urich, after the extension on his sick wife's hospital stay is denied and he is offered a better paying job as an editor rather than a reporter, decides to give up the investigation. Page, in an attempt to change his mind, takes him to a rest home she just discovered, where they meet Ma rlene Vistain, Fisk's mother, who tells them that Fisk killed his father. Fisk, meanwhile, is thro wing a charity gala to publicly raise money for victims of the bombings. There, many of the guests , including Marianna, are poisoned.Still recovering from his injuries, Murdock says goodbye to Tem ple, who is leaving for a time. Before she goes, Temple suggests that he get better protection if he is going to continue his crusade. With the help of Barrett, Murdock finds Melvin Potter, a ment ally unbalanced engineer who has been coerced into creating armored clothing for Fisk, and asks hi m to make a suit of body armor in exchange for stopping Fisk from hurting anyone else. Page wishes to reveal the story of how Vistain is alive, and says that Fisk killed his father, but Urich expla ins that it is unreliable, given her state of mind. While Fisk will not leave Marianna's side as she recovers in hospital, Wesley receives a phone call from Vistain and learns that Page and Urich visited her. He confronts Page and attempts to blackmail her into not exposing Fisk, threatening t o hurt her friends. When Fisk calls Wesley to find out where he is, the ringing distracts him long enough for Page to take his gun, and kill him.Using information from Urich, Murdock finds the base of Gao's heroin operation and dismantles it, with Gao, who can hold her own against Murdock, escap ing and deciding to return to her homeland to think about the future. Fisk's men find Wesley, and the grieving Fisk realizes that the last person he talked to was Vistain. Wanting to keep those he loves safe by sending them out of the country, Fisk is unable to convince Marianna to leave him, b ut does get Vistain away, with the sick Vistain unable to tell him what her call to Wesley was abo ut. Nelson continues his work without Murdock, and takes what they know to Stahl. Page, struggling to get over killing Wesley, convinces Urich to write the story, but it is rejected by Ellison. Whe n Urich accuses Ellison of being paid off by Fisk, he gets fired. Urich decides to start his own blog to get Fisk's story out there, but Fisk's actual informant in the Bulletin tells him that Uri ch visited Vistain, angering him to the point that he breaks into Urich's apartment and kills him. Murdock and Page attend Urich's funeral, while Fisk learns that Owlsley and Gao conspired to poiso n Marianna, whom they deemed a distraction, and that Owlsley has been hiding Hoffman as an insurance policy. When Owlsley tries to blackmail him, an infuriated Fisk kills Owlsley by throwing him into an open elevator shaft. Murdock and Nelson reconcile their differences, and begi n building their case against Fisk. As the vigilante, Murdock finds Hoffman with information from Stahl and convinces him to testify against Fisk. Acting on this testimony, federal agents arrest F isk and his co-conspirators, but Fisk manages to escape custody. Before he can flee the city, Murd ock intercepts him, wearing the new armor made for him by Potter. After a brutal fight, Murdock de feats Fisk and leaves him to the police. With Fisk arrested, Marianna leaves the city. Murdock, Pa ge, and Nelson celebrate their success and resume their work. The vigilante is named "Daredevil" b y the media.Season 2In the aftermath of Wilson Fisk's arrest, the firm of Nelson & Murdock enjoys newfound fame within Hell's Kitchen, but only attracts few new clients who can afford to pay their legal fees. While Karen Page manages their financial troubles, Foggy and Matt struggle with Daredevil's surge in popularity and the increased danger to their partnership, firm, and loved ones. Meanwhile, a new assailant arrives in Hell's Kitchen, systematically eliminating rival gangs in their home territory. After the only survivor, Grotto, hires Nelson & Murdock to represent him, Daredevil finds himself in a deadly confrontation with the vigilante and is shot off a building at point blank range.After being severely injured during his fight with the unknown assailant, Foggy insists that Matt recovers before trying to track him down again. Matt enlists the help of an old ally to rebuild the broken pieces of his suit while Foggy and Karen tend to Grotto, who is targeted by the assailant who wants to finish the job. In a deal with District Attorney Reyes, Fog gy and Karen secure a witness protection deal in exchange for Grotto procuring incriminating infor mation on a higher-level drug lord, but are double-crossed when Reyes instead uses him as bait for the newly-dubbed "Punisher." During an intense battle between Daredevil and the Punisher, Law Enfo rcent begin firing recklessly at both of them under Reyes' orders and in the aftermath the two are nowhere to be found.Now captured by the Punisher, Matt engages him in a debate about their vigilante methods while trying to plan an escape. Foggy and Karen deal with the aftermath of Reyes' trap, with Foggy searching for Matt and Karen pursuing their missing client. As the Punisher (now identified as "Frank") prepares to intensify his attacks on local gangs, Matt engages him in a series of heated arguments about the morality of their actions, culminating in ea ch questioning and reaffirming their motives. Their debate culminates in Frank strapping a gun wit h a single bullet to Matt's hand and offering Daredevil a choice: kill him before he kills Grotto, or do nothing so that he can get on with his "job." Matt escapes but cannot save Grotto; at the sa me time, Frank executes his next attack on a local gang, with the members coming to kill them both . Matt captures Frank but he loses him when he is forced to fight off a swarm of attackers.Finn Co oley, a high-profile member of the Irish mafia whose son was killed by the Punisher, sends his rem aining associates in search of him and the money he stole from the mafia. Elsewhere, Karen learns from case files provided by Reyes subordinate, Blake Tower, that the Punisher's real name is Frank Castle, and locates a disgraced surgeon who points her to the Castle household. That night, Castle is captured by the Irish and taken off to be brutally tortured by Cooley who drills through his fo ot. He gives up the location of the money when Cooley threatens to kill Castle's dog, but the case containing the money is armed with a detonated bomb that kills some of Cooley's men. Cooley prepares to kill Castle, who escapes and kills Cooley and several henchmen. At the same time, Matt arrives at the site and assists a wounded Castle in incapacitating Cooley's remaining henchmen, be fore weakly carrying him to a nearby cemetery. There, Castle tells Matt about his family, which he

reveals was murdered by the mafia. The police arrive and apprehend Castle, and Matt tells Sergeant Mahoney to take credit for the arrest to reinforce the city's hope in the police force. Matt, Foggy and Karen celebrate at Josie's Bar, and Matt and Karen share a kiss in the rain before parting ways. When Matt returns home, he is surprised by the appearance of an old friend, Elektra Natchios.10 years ago, Matt meets Elektra at a ballroom party. The two bond over their shared observational skills and thrill seeking before heading off into the night in a stolen car. They visit the boxing ring used by Matt's late father, where Matt discloses details about his father's death before the two spar and begin having sex midway. Sometime later, they break into the home of Roscoe Sweeney, the man responsible for ordering the death of Matt's father. Sweeney is restrained, and Matt beats him unconscious. Elektra urges Matt to kill Sweeney, but Matt is unable to, which prompts Elektra to abandon him. In the present day, Elektra requests Matt's help on her dealings with her father's company Roxxon Energy Corporation, which Matt refuses, though he spies on a business meeting. That night, he and Karen go on their first date together. Matt visits Elektra's penthouse, where Yakuza members begin to converge on the building. Elektra, who has stolen Matt's costume, advises him to prepare to fight.Matt and Elektra subdue Yakuza assailants that attack Elektra's apartment. Afterwards, they talk in a diner and Matt reluctantly agrees to continue helping her. Matt, Foggy and Karen meet with the public defender on the Punisher case, who reveals that Castle could be facing the death penalty for links to crimes outside New York. This prompts Matt to consider defending Castle himself. Their meeting at the hospital is interrupted by Reyes, who is insisting on giving Castle the death penalty. She is told off by Matt, but his involvement in the case is cut short when Elektra recruits him to steal a valuable Yakuza ledger from the Roxxon building. While the two succeed in their heist and narrowly escape, Karen reveals to Castle that she broke into his home and discovered private details about his family. The two quietly converse over this until Foggy comes in and informs Castle that he's managed to shorten the latter's charges to one life sentence, as long as he pleads guilty. Castle agrees, but when stating his plea to Reyes and the judge, he suddenly pleads not guilty and threatens Reyes. When Matt returns to the firm, Foggy informs him The People v. Frank Castle has begun.Castle's trial begins. Matt neglects his work on Castle's case to continue moonlighting with Elektra. The two then raid a train car and get into an extended fight with the Yakuza, only for Elektra to be wounded. After, Matt tends to her wounds and asks why she left him, only to be told he "deserves better". The next morning, Matt is late to make his opening statement at Castle's trial and a reluctant Foggy is forced to take over for him. Matt's relationship with Karen and Foggy now continues to deteriorate. Matt reveals his work with Elektra to Foggy and Foggy, who he assumes is having an affair with her, suggest Matt leaves the case. That night, Matt and Elektra force a local corrupt professor to decrypt the yakuza's ledger and it leads them to an abandoned warehouse where they discover the Yakuza are digging a giant hole.Matt and Elektra are attacked by ninjas and saved by Stick. Elektra is cut by a poisoned sword and rushed back to Matt's apartment so Stick can save her. Stick reveals that Elektra works for him, then tells Matt about The Hand and their purpose in New York. In court, Foggy begins to sway the jury in Castle's favor. Matt offers to take Elektra back if she'll leave Stick and swear not to kill any more. Karen comes by and is devastated to see a recovering Elektra in Matt's bed. In court the next day, Frank takes the stand and purposefully wrecks his own defense. Karen and Foggy berate Matt for allowing them to lose. Elektra tells Stick that she's leaving him to be with Matt; Matt and her share a close moment before they're attacked by an assassin. They subdue him and discover that he's just a kid, only Elektra impulsively slits his throat much to Matt's horror. Castle enters prison and is led by a guard into a meeting with Wilson Fisk.Fisk arranges a meeting with Castle, who reluctantly agrees to a deal which involves killing Fisk's rival in the prison who is able to provide information on the massacre of his family. Betrayed by Fisk, Castle single highhandedly slaughters a swarm of henchmen of the man he had just killed and is placed in solitary. After an intense brawl with Fisk, he is smuggled out of prison, now able to find the mastermind behind his family's massacre named the Blacksmith, while promising Fisk that he will kill him the next time they see each other. Matt and Elektra fall out for good over Elektra's indifference to killing. Foggy and Matt reluctantly agree to part ways after Nelson & Murdock collapses. Karen confirms the John Doe at the carousel was really an undercover cop, knowledge Castle learned from his cell block victim. Matt as Daredevil learns the accountant's son is one of several children held for a medical experiment by Nobu, the now resurrected Yakuza leader. Nobu manages to escape with a device to which the children were hooked. Matt wonders whether Stick's claim that the Hand has discovered immortality can be true.Matt as Daredevil arranges hospital treatment for the children who'd been hooked up to Nobu's blood extractor. Reyes brings Foggy & Matt to her office to get information on Castle that may keep her family alive. Foggy & Matt won't break attorney-client privilege, forcing Reyes to tell the whole story of the carousel massacre---a major drug deal gone awry as Blacksmith, its mastermind, failed to show when learning of the undercover cop, and tensions spilling into the gun fight that killed the cop and Castle's family. Reyes admits to covering the entire matter up for fear of career ruin---including threatening the medical examiner and approving Castle's do-not-resuscitate. Before she can divulge more details, a firestorm of bullets riddles the office. Reyes is killed and Foggy is injured. Matt learns Castle was taken to Fisk's cellblock and visits Fisk, a visit ending in a fight when Matt unsuccessfully blackmails Fisk into revealing his involvement by assuring he will never see Vanessa again. Karen learns the hard way that Castle may not have been involved in the attack on Reyes's office---a similar attack happens at her apartment while Castle himself visits and saves her. Elektra kills a charming assassin sent to kill her---by Stick. Matt on the hospital roof prepares to confront a horde of the Hand who might attack either Foggy or the recovering children.Daredevil takes on the ninjas assaulting the hospital. He's shocked that the children want to return to them. As Matt, he explains to Claire Temple about the ninjas. An attempted autopsy on one of the slain Ninjas shows the scars of a prior autopsy. The hospital board chief is just as shocked---and bent on covering everything up, compelling Claire to quit after Foggy's release. Foggy ponders an offer from his occasional girlfriend involving joining her law firm. Karen lies to a detective about the attack on her apartment but accepts police protection. Matt confronts her and she tells him she doesn't think Castle was behind the D.A.'s

and medical examiner's deaths. Karen slips her police protection to meet Castle at a greasy spoon. Castle admits he used her as bait to lure those he thinks are following her. Matt thinks Madame Ga o is Blacksmith's competitor and she reluctantly sends him to the pier. He talks Castle out of kil ling a man claiming to be Blacksmith. Stick learns Elektra survived the hit. Blacksmith's men arri ve and shoot up the boat, detonating gunpowder and exploding the ship.Matt stops Elektra from kill ing Stick as ninjas descend on them and they must fight together, but Matt and Elektra can't stop the Hand from taking Stick. Matt vows to find him, while Elektra vows to kill him. Police clean up the pier mess and Karen insists Castle isn't dead. She tracks down Castle's commander in Afghanistan, Ray Schoonover, then discovers Schoonover's involvement in the drug ring that led to the slaughter of Castle's family. Castle shows up to keep Schoonover from killing Karen. Daredevil and Elektra track the Hand and find Stick bound and tortured. Nobu is also present. The Hand's tru e motive is disclosed, as is the missing link in Elektra's haunted past. Elektra finally saves Stick. Daredevil escapes the Hand with Nobu determined to stop him.Foggy meets with Jeri Hogarth t o discuss joining her firm. With Stick safe at Matt's apartment, the Hand raids a police station f or information on people Daredevil has helped. When Matt learns of it, he fears especially for Kar en. Matt and Elektra use a police transmission to find the beneficiaries the Hand rounded up. The hostages are freed, exposing a trap intended for Daredevil. Matt and Elektra make a pact to stay t ogether after defeating the Hand, even leaving New York. Nobu arrives and attacks Daredevil, but E lektra sacrifices herself for him. An enraged Daredevil takes on the remaining ninjas with unexpected help---from Castle. Stick returns to finish off Nobu, then mourns Elektra with Matt. Foggy accepts the offer from Hogarth's firm, formally ending Nelson & Murdock, and has a final dri nk with Karen before a final talk with Matt. Castle burns down his family's home, after taking a C D labelled "Micro". Matt reveals to Karen that he is Daredevil. The Hand recovers Elektra's body.
==========
A man (Robert Mitchum) enters a courthouse and asks for Sam Bowden. He watches Sam (Gregory Peck) in a trial, then follows him to his car, re-introducing himself as Max Cady. Max makes vaguely thr eatening comments as Sam drives off. Later, Max watches Sam and his family bowling, and makes a pa ss at the waitress. Sam calls police chief Dutton (Martin Balsam) and tells him that Max is being creepy. He explains that Max is an ex-convict who may have a grudge against Sam, and that Sam's fa mily may be at risk. Dutton calls his deputies and tells them to pick up Max for vagrancy. Max coo perates and obviously knows the law, so they release him.The next day, Sam finds their dog, poisoned. Sam suspects Max but cannot prove it. His wife Peggy (Polly Bergen) and daughter Nancy (Lori Martin) are distraught, and Sam explains about Max.Max hires a fiery lawyer, Dave Grafton (Ja ck Kruschen), who accuses the police of harassing Max. Dutton tells Sam he cannot do anything unti l Max commits a crime, but advises Sam to hire private detective Charles Sievers. Sievers (Telly S avalas) starts following Max. He calls the police to have Max picked up for lewd vagrancy, but by the time they arrive, Max is gone, and the woman he was with (Barrie Chase) is beaten up. She refu ses to help them and leaves town, fearful of Max's revenge. Sievers suggests Sam hire a dock thug, but he refuses.The next day, Sam catches Max leering at his daughter and takes a swing at him. Max doesn't punch back. Coming out from school, Nancy sees Max and flees, eventually running in front of a car. She's ok, but anger builds in Sam, and he gets a gun. Peggy convinces him not to go afte r Max.Sam meets Max in a bar and tries to pay him off. Max says he prefers "death by a thousand cu ts", and turns him down. Sam hires the thugs to beat up Max, but he fights them off. Max calls Sam and tells him that he has the law on his side. Sam and Peggy discuss a plan to use Nancy as bait. San tells Dutton that he will "hide" his family in a boat on the Cape Fear River, then asks for a deputy to help him catch Max.Sam and Deputy Kersek (Page Slattery) stake out the boat. Max arrives and sneaks up on Kersek, killing him. He sets the boat with Peggy on it adrift, and Sam chases aft er. Nancy calls the sheriff from the dock. Max attacks Peggy, but leaves for Nancy when Sam arrive s. Sam and Max fight; Sam shoots Max, and Max begs Sam to finish him off. Sam declines, saying Max will rot in jail. The family leaves together.

In [23]:

```
# https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

In [24]:

```
# https://gist.github.com/sebleier/554280
```

```
# we are removing the words from the stop words list: 'no', 'nor', 'not'
# <br /><br /> ==> after the above steps, we are getting "br br"
# we are including them into stop words list
# instead of <br /> if we have <br/> these tags would have revmoved in the 1st step

stopwords= set(['br', 'the', 'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "y
ou're", "you've",\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his',
'himself','not' \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll",
'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', '
while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',
'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under'
, 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'e
ach', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll'
, 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "d
esn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn',
"mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn',
"wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"])
```

In [25]:

```
# Combining all the above stundents
from tqdm import tqdm
from bs4 import BeautifulSoup

preprocessed_synopsis = []
# tqdm is for printing the status bar
for sentance in tqdm(clean_data['plot_synopsis'].values):
    sentance = re.sub(r"http\S+", "", sentance)
    sentance = BeautifulSoup(sentance, 'lxml').get_text()
    sentance = decontracted(sentance)
    sentance = re.sub("\S*\d\S*", "", sentance).strip()
    sentance = re.sub('[^A-Za-z]+', ' ', sentance)
    # https://gist.github.com/sebleier/554280
    sentance = ' '.join(e.lower() for e in sentance.split() if e.lower() not in stopwords)
    preprocessed_synopsis.append(sentance.strip())
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 14752/14752 [01:
23<00:00, 176.98it/s]
```

In [26]:

```
preprocessed_synopsis[0]
```

Out[26]:

```
'note synopsis orginal italian release segments certain order boris karloff introduces three
horror tales macabre supernatural known three faces fear telephonerosy michele mercier attractive h
igh priced parisian call girl returns spacious basement apartment evening she immediately gets bes
et series strange phone calls caller soon identified frank ex pimp recently escaped prison rosy te
rrified testimony landed man jail looking solace rosy phones lesbian lover mary lynda alfonsi two
women estranged time rosy certain she one help mary agrees come night seconds later frank calls pr
omising no matter she calls protection revenge unknown rosy mary caller impersonating frank marry
arrives rosy apartment soon best calm rosy nerves she gives panic struck woman tranquillizer puts
bed later night rosy sleeps mary gets bed pens note confession she one making strange phone calls
she learned franks escape prison knowing rosy would call help she explains she felt way coming bac
k life breakup she busy writing she fails notice intruder apartment time frank real creeps behind
mary strangles death one rosys nylon stockings sound struggle awaken rosy she gasps fright
murderous pimp realizes killed wrong woman slowly makes way rosy bed however earlier night rosy pl
```

```
aced butcher knife pillow mary suggestion rosy seizes knife stabs frank beginning strangle rosy dr
ops knife breaks hysteria surrounded two corpses former lovers wurdalakin century russia vladimir
urfe young nobleman long trip course journey finds beheaded corpse knife plunged heart withdraws b
lade takes souvenir later night vladimir stops small rural cottage ask shelter notices several
daggers hanging one walls vacant space happens fit one discovered vladimir surprised entrance gior
gio glauco onorato explains knife belongs father not seen five days giorgio offers room young coun
t subsequently introduces rest family wife rika dialina young son ivan giorgio younger brother
pietro massimo righi sister sdenka susy anderson subsequently transpires eagerly anticipating
arrival father gorcha well reason absence gone battle outlaw dreaded wurdalak ali beg vladimir con
fused term sdenka explains wurdalak walking cadaver feeds blood living preferably close friends fa
mily members giorgio pietro certain corpse vladimir discovered ali beg also realize strong possibi
lity father infected blood curse warn count leave decides stay await old mans return stroke
midnight gorcha boris karloff returns cottage sour demeanor unkempt appearance bode worse two
brothers torn realize duty kill gorcha feeds family love makes difficult reach decision later
night ivan pietro attacked gorcha drains blood flees cottage giorgio stakes beheads pietro prevent
reviving wurdalak prevented ivan wife threatens commit suicide reluntantly agrees bury child witho
ut taking necessary precautions night child rises grave begs invited cottage mother runs son aid s
tabbing giorgio attempts stop greeted front door gorcha old man bits infects daughter law husband
vladimir sdenka flee cottage go run hide ruins abandoned cathedral dawn breaks vladimir optimistic
long happy life lies sdenka reluctant relinquish family ties she believes she meant stay family sd
enka fears family confirmed evening gorcha siblings show abandoned abby vladimir sleeps sdenka lur
ed loving arms bite death awakened screams vladimir rushes aid family already taken home forcing l
over follow suite young nobleman finds lying motionless bed sdenka awakens distinct change visible
face no longer caring vladimir embraces she bites infects drop waterin victorian london england nu
rse helen chester jacqueline pierreux called large house prepare corpse elderly medium burial she
dressed body she notices elaborate diamond ring finger tempted greed nurse chester steals she glas
s tips drops water begin splash floor she also assailed fly no doubt attracted odor body unsettled
pleased acquisition she finishes job returns home small east end flat returning home nurse chester
assailed strange events buzzing fly returns continues pester lights apartment go sounds dripping w
ater continues maddening regularity she sees old womans corpse lying bed coming towards terrified
woman begs forgiveness she ultimately strangles imaging medium hands gripping throat next morning
concierge harriet white medin discovers nurse chester body calls police investigator scene gustavo
de nardo quickly concludes simple case nurse chester died fright pathologist arrives scene examine
body taken away notes sign violence small bruise left finger mostly likely caused someone pried ri
ng finger doctor makes observation concierge appears distressed she apparently took ring dead nurs
e chester distracted sound fly swooping air boris karloff makes final appearance gorcha riding
horse concludes three tales fear tells viewers careful walking home night ghosts vampires no fear
image pulls back actually reveal sitting prop fake horse camera crew various crewmen moving
branches around simulate scene riding forest wurdalak segment'
```

In [7]:

```
clean_data['cleaned_synopsis'] = preprocessed_synopsis
```

4. Machine Learning Models

4.1.1 Feauture Engineering

From EDA we have observed that the tags are unbalanced and most of the tags are 1 . we will consder 3 tags and 5 tags per synopsis As We have Observed that avg tags per synposis are 3.

4.1.2 Converting tags for multilabel problems

X	y1	y2	y3	y4
x1	0	1	1	0
x1	1	0	0	0
x1	0	1	0	0

5 Tags per synopsis

In [8]:

```
# binary='true' will give a binary vectorizer
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(tokenizer = lambda x: x.split(','), binary='true',max_features = 5)
multilabel_y_5 = vectorizer.fit_transform(clean_data['tags'])
```

3 Tags per synopsis

In [9]:

```
# binary='true' will give a binary vectorizer
from sklearn.feature_extraction.text import CountVectorizer
vectorizer1 = CountVectorizer(tokenizer = lambda x: x.split(','), binary='true',max_features = 3)
multilabel_y_3 = vectorizer1.fit_transform(clean_data['tags'])
```

4.2 Split the data into test,CV and train

5 Tags per synopsis

In [10]:

```
from sklearn.model_selection import train_test_split

x,x_test,y,y_test =  train_test_split(clean_data['cleaned_synopsis'],multilabel_y_5,train_size=0.8)
x_train,x_cv,y_tain,y_cv =  train_test_split(x,y,train_size=0.8)
```

3 Tags per synopsis

In [11]:

```
from sklearn.model_selection import train_test_split

x1,x_test1,y1,y_test1 =  train_test_split(clean_data['cleaned_synopsis'],multilabel_y_3,train_size
=0.8)
x_train1,x_cv1,y_tain1,y_cv1 =  train_test_split(x1,y1,train_size=0.8)
```

4.3 Featurizing data

We will be using below features for predictiong the tags

1) Bag of Words
2) Ngrams
3) UniGrams
4) BiGrams
5) TriGrams
6) Char-3 Grams
7) Char-5 Grams
8) Uni+BI+Tri Grams
9) Char-3+char-5

Refer :https://www.aclweb.org/anthology/L18-1274 Section 4.1

4.3.1 Bag Of Words

5 Tags per synopsis

In [12]:

```
from sklearn.feature_extraction.text import CountVectorizer

#bow = CountVectorizer(ngram_range=(1,3),max_features=100000)
bow = CountVectorizer()
x_train_bag1 = bow.fit_transform(x_train)
x_cv_bag1 = bow.transform(x_cv)
x_test_bag1 = bow.transform(x_test)
```

3 Tags per synopsis

In [13]:

```
from sklearn.feature_extraction.text import CountVectorizer

#bow = CountVectorizer(ngram_range=(1,3),max_features=100000)
bow = CountVectorizer()
x_train_bag_1 = bow.fit_transform(x_train1)
x_cv_bag_1 = bow.transform(x_cv1)
x_test_bag_1 = bow.transform(x_test1)
```

4.3.2 Ngrams

5 Tags per synopsis

In [14]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

#tfidf = TfidfVectorizer(max_features=10000,sublinear_tf=False)
tfidf = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_idf=True, norm="12",tokenizer =
lambda x: x.split(), sublinear_tf=False, ngram_range=(1,3))
x_train_tfidf = tfidf.fit_transform(x_train)
x_cv_tfidf = tfidf.transform(x_cv)
x_test_tfidf = tfidf.transform(x_test)
```

3 Tags per synopsis

In [15]:

```
#tfidf = TfidfVectorizer(max_features=10000,sublinear_tf=False)
tfidf = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_idf=True, norm="12",tokenizer =
lambda x: x.split(), sublinear_tf=False, ngram_range=(1,3))
x_train_tfidf_th = tfidf.fit_transform(x_train1)
x_cv_tfidf_th = tfidf.transform(x_cv1)
x_test_tfidf_th = tfidf.transform(x_test1)
```

4.3.3 Uni Gram

5 Tags per synopsis

In [16]:

```
# unigram for plot_synopsis
from datetime import datetime
start = datetime.now()
vectorizer = TfidfVectorizer(min_df=0.00009, max_features=20000, smooth_idf=True, norm="12", \
                             tokenizer = lambda x: x.split(), sublinear_tf=False, ngram_range=(1,1))
x_train_tfidf1 = vectorizer.fit_transform(x_train)

x_cv_tfidf1 = vectorizer.transform(x_cv)
x_test_tfidf1 = vectorizer.transform(x_test)
print("Time taken to run this cell :", datetime.now() - start)
```

```
Time taken to run this cell : 0:00:05.025344
```

3 Tags per synopsis

In [17]:

```
# unigram for plot_synopsis
from datetime import datetime
start = datetime.now()
vectorizer = TfidfVectorizer(min_df=10, max_features=20000, smooth_idf=True, norm="12", \
                             tokenizer = lambda x: x.split(), sublinear_tf=False, ngram_range=(1,1))
x_train_tfidf_th1 = vectorizer.fit_transform(x_train1)

x_cv_tfidf_th1 = vectorizer.transform(x_cv1)
```

```
x_test_tfidf_th1 = vectorizer.transform(x_test1)
print("Time taken to run this cell :", datetime.now() - start)
```

```
Time taken to run this cell : 0:00:05.597166
```

4.3.4 Bi Gram

5 Tags per synopsis

In [18]:

```
#tfidf = TfidfVectorizer(max_features=10000,sublinear_tf=False)
tfidf = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_idf=True, norm="l2",tokenizer =
lambda x: x.split(), sublinear_tf=False, ngram_range=(2,2))
x_train_tfidf2 = tfidf.fit_transform(x_train)
x_cv_tfidf2 = tfidf.transform(x_cv)
x_test_tfidf2 = tfidf.transform(x_test)
```

3 Tags per synopsis

In [19]:

```
#tfidf = TfidfVectorizer(max_features=10000,sublinear_tf=False)
tfidf = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_idf=True, norm="l2",tokenizer =
lambda x: x.split(), sublinear_tf=False, ngram_range=(2,2))
x_train_tfidf_th2 = tfidf.fit_transform(x_train1)
x_cv_tfidf_th2 = tfidf.transform(x_cv1)
x_test_tfidf_th2 = tfidf.transform(x_test1)
```

4.3.5 Tri Gram

5 Tags per synopsis

In [27]:

```
#tfidf = TfidfVectorizer(max_features=10000,sublinear_tf=False)
tfidf3 = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_idf=True, norm="l2",tokenizer
= lambda x: x.split(), sublinear_tf=False, ngram_range=(3,3))
x_train_tfidf3 = tfidf3.fit_transform(x_train)
x_cv_tfidf3 = tfidf3.transform(x_cv)
x_test_tfidf3 = tfidf3.transform(x_test)
```

3 Tags per synopsis

In [20]:

```
#tfidf = TfidfVectorizer(max_features=10000,sublinear_tf=False)
tfidf3 = TfidfVectorizer(min_df=0.00009, max_features=200000, smooth_idf=True, norm="l2",tokenizer
= lambda x: x.split(), sublinear_tf=False, ngram_range=(3,3))
x_train_tfidf_th3 = tfidf3.fit_transform(x_train1)
x_cv_tfidf_th3 = tfidf3.transform(x_cv1)
x_test_tfidf_th3 = tfidf3.transform(x_test1)
```

4.3.6 Char-3 gram

5 Tags per synopsis

In [21]:

```
tfidfchar3 = TfidfVectorizer(sublinear_tf=True, strip_accents='unicode', analyzer='char',
ngram_range=(3, 3),  max_features=20000)
x_train_tfidf4 = tfidfchar3.fit_transform(x_train)
x_cv_tfidf4 = tfidfchar3.transform(x_cv)
```

```
x_test_tfidf4 = tfidfchar3.transform(x_test)
```

3 Tags per synopsis

In [22]:

```
tfidfchar3 = TfidfVectorizer(sublinear_tf=True, strip_accents='unicode', analyzer='char',
ngram_range=(3, 3),  max_features=20000)
x_train_tfidf_th4 = tfidfchar3.fit_transform(x_train1)
x_cv_tfidf_th4 = tfidfchar3.transform(x_cv1)
x_test_tfidf_th4 = tfidfchar3.transform(x_test1)
```

4.3.7 Char-5 Gram

5 Tags per synopsis

In [23]:

```
tfidfchar5 = TfidfVectorizer(sublinear_tf=True, strip_accents='unicode', analyzer='char',
ngram_range=(5, 5),  max_features=20000)
x_train_tfidf5 = tfidfchar5.fit_transform(x_train)
x_cv_tfidf5 = tfidfchar5.transform(x_cv)
x_test_tfidf5 = tfidfchar5.transform(x_test)
```

3 Tags per synopsis

In [24]:

```
tfidfchar5 = TfidfVectorizer(sublinear_tf=True, strip_accents='unicode', analyzer='char',
ngram_range=(5, 5),  max_features=20000)
x_train_tfidf_th5 = tfidfchar5.fit_transform(x_train1)
x_cv_tfidf_th5 = tfidfchar5.transform(x_cv1)
x_test_tfidf_th5 = tfidfchar5.transform(x_test1)
```

4.3.8 Uni+Bi+Tri Grams

5 Tags per synopsis

In [25]:

```
from scipy.sparse import coo_matrix, hstack
xtrain_train_tfidf12 = hstack((x_train_tfidf1,x_train_tfidf2),format="csr",dtype='float64')
xtrain_test_tfidf12 = hstack((x_test_tfidf1,x_test_tfidf2),format="csr",dtype='float64')
xtrain_cv_tfidf12 = hstack((x_cv_tfidf1,x_cv_tfidf2),format="csr",dtype='float64')
```

In [28]:

```
x_train_tfidf123 = hstack((xtrain_train_tfidf12,x_train_tfidf3),format="csr",dtype='float64')
x_test_tfidf123 = hstack((xtrain_test_tfidf12,x_test_tfidf3),format="csr",dtype='float64')
x_cv_tfidf123 = hstack((xtrain_cv_tfidf12,x_cv_tfidf3),format="csr",dtype='float64')
```

3 Tags per synopsis

In [29]:

```
x_train_tfidf_th12 = hstack((x_train_tfidf_th1,x_train_tfidf_th2),format="csr",dtype='float64')
x_test_tfidf_th12 = hstack((x_test_tfidf_th1,x_test_tfidf_th2),format="csr",dtype='float64')
x_cv_tfidf_th12 = hstack((x_cv_tfidf_th1,x_cv_tfidf_th2),format="csr",dtype='float64')
x_train_tfidf_th123 = hstack((x_train_tfidf_th12,x_train_tfidf_th3),format="csr",dtype='float64')
x_test_tfidf_th123 = hstack((x_test_tfidf_th12,x_test_tfidf_th3),format="csr",dtype='float64')
x_cv_tfidf_th123 = hstack((x_cv_tfidf_th12,x_cv_tfidf_th3),format="csr",dtype='float64')
```

4.3.9 Char-3 + Char-5

5 Tags per synopsis

In [30]:

```
from scipy.sparse import coo_matrix, hstack
xtrain_train_tfidf45 = hstack((x_train_tfidf4,x_train_tfidf5),format="csr",dtype='float64')
xtrain_test_tfidf45 = hstack((x_test_tfidf4,x_test_tfidf5),format="csr",dtype='float64')
xtrain_cv_tfidf45 = hstack((x_cv_tfidf4,x_cv_tfidf5),format="csr",dtype='float64')
```

3 Tags per synopsis

In [31]:

```
x_train_tfidf_th45 = hstack((x_train_tfidf_th4,x_train_tfidf_th5),format="csr",dtype='float64')
x_test_tfidf_th45 = hstack((x_test_tfidf_th4,x_test_tfidf_th5),format="csr",dtype='float64')
x_cv_tfidf_th45 = hstack((x_cv_tfidf_th4,x_cv_tfidf_th5),format="csr",dtype='float64')
```

4.4 Applying SVM with OneVsRest Classifier

Defining the function to train with SVM

In [47]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import
log_loss,accuracy_score,confusion_matrix,classification_report,f1_score,hamming_loss
import seaborn as sns
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
def svm(x_train,y_train,x_cv,y_cv):

    alpha = [10**x for x in range(-4,4)]

    cv_log_error =[]
    train_log_error=[]
    for i in alpha:

        clf_dt = OneVsRestClassifier(SGDClassifier(loss='hinge',alpha=i,penalty='l2',class_weight='
balanced'))

        clf_dt.fit(x_train,y_train)

        predict_ytrain = clf_dt.predict(x_train)

        predict_y_cv = clf_dt.predict(x_cv)

        f1score_train = f1_score(y_train,predict_ytrain,average='micro')
        f1score_cv = f1_score(y_cv,predict_y_cv,average='micro')
        train_log_error.append(f1score_train)
        print('The f1 score of cv data for alpa ',i,'is',f1score_cv)
        cv_log_error.append(f1score_cv)
    fig, ax = plt.subplots(figsize=(20,7))
    plt.xscale('log')
    ax.plot(alpha, cv_log_error,c='g',label='CV_f1 score')
    for i, txt in enumerate(np.round(cv_log_error,3)):
        ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],cv_log_error[i]))
    ax.plot(alpha, train_log_error,c='b',label='Train_f1 score')
    for i, txt in enumerate(np.round(train_log_error,3)):
        ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],train_log_error[i]))

    plt.grid()
    plt.title("F1 score for each alpha ")
    plt.xlabel("alpha values's",)
    plt.ylabel("f1 score")

    plt.legend()
    plt.show()
```

Defining svm function for best parameters

In [27]:

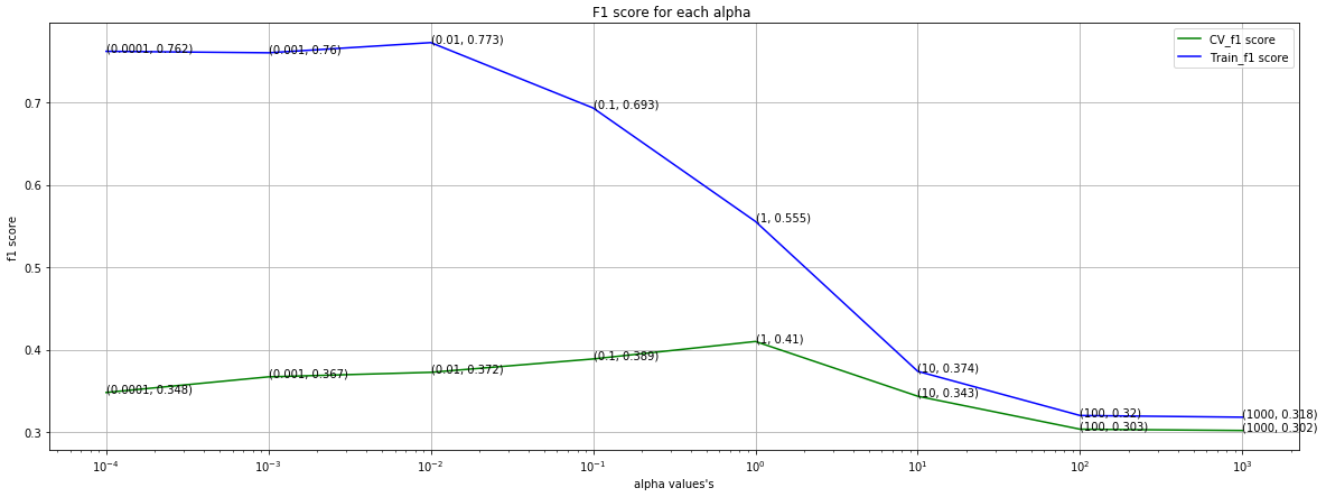
```
def best_svm_linear(x_train,y_train,x_test,y_test,c):



        clf_dt_best = OneVsRestClassifier(SGDClassifier(alpha=c,loss='hinge',penalty='l2',class_wei
ght='balanced'))
        clf_dt_best.fit(x_train, y_train)
        predict_y_train = clf_dt_best.predict(x_train)
        print('For values of c = ', c, "The train f1  score is:",f1_score(y_train,predict_y_train,a
verage='micro'))
        predict_y_test = clf_dt_best.predict(x_test)
        print('For values of  c = ', c, "The test  f1 score is:",f1_score(y_test,predict_y_test,ave
rage='micro'))

        acc = hamming_loss(y_test,predict_y_test)
        print('Hamming loass on test data is ',acc)


        print("Precision recall report :\n",classification_report(y_test, predict_y_test))
```

4.4.1 Bag of words with 5 tags

In [54]:

```
svm(x_train_bag1,y_tain,x_cv_bag1,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.34789987271955874
The f1 score of cv data for alpa  0.001 is 0.36687277496721005
The f1 score of cv data for alpa  0.01 is 0.37235437347817946
The f1 score of cv data for alpa  0.1 is 0.38857915188376413
The f1 score of cv data for alpa  1 is 0.409814126394052
The f1 score of cv data for alpa  10 is 0.343348468746981
The f1 score of cv data for alpa  100 is 0.3033208402335919
The f1 score of cv data for alpa  1000 is 0.30174825174825176
```



we can see from above plot we can say alpha=1 we are getting better results

In [55]:

```
best_svm_linear(x_train_bag1,y_tain,x_test_bag1,y_test,1)
```

```
For values of c =  1 The train f1  score is: 0.5413639213661343
For values of  c =  1 The test  f1 score is: 0.4049040970192874
```

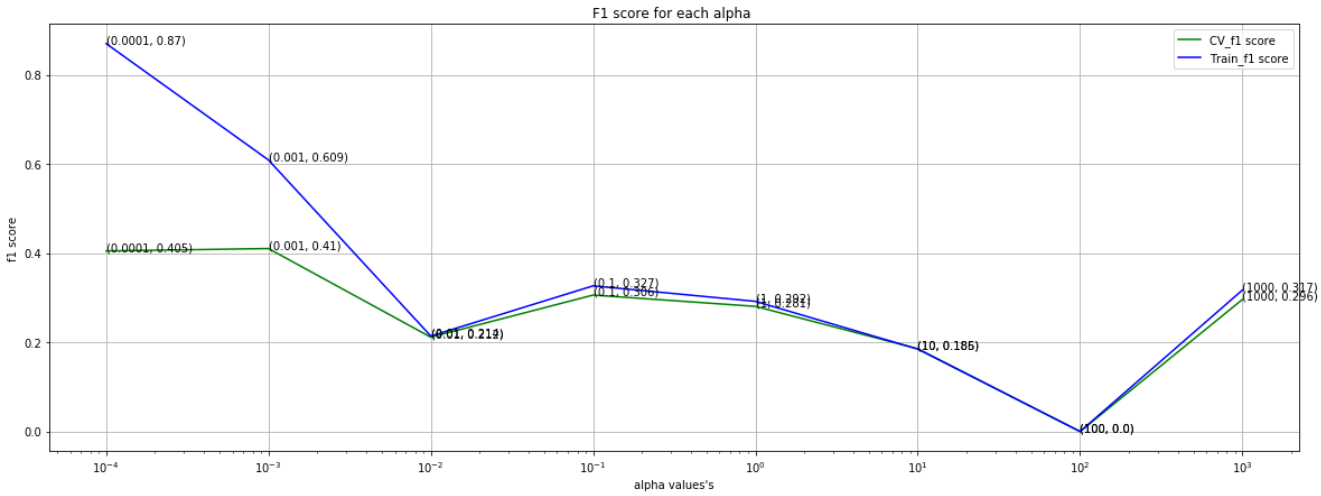
```
For values of  c =  1 The test  f1 score is: 0.4049094097019274
Hamming loass on test data is  0.3450355811589292
Precision recall report :
              precision    recall  f1-score   support

           0       0.18      0.62      0.28       320
           1       0.24      0.68      0.35       522
           2       0.48      0.74      0.58       871
           3       0.35      0.64      0.45       573
           4       0.17      0.55      0.26       305

   micro avg       0.29      0.67      0.40      2591
   macro avg       0.28      0.65      0.39      2591
weighted avg       0.33      0.67      0.43      2591
 samples avg       0.21      0.34      0.24      2591
```

4.4.2 Ngrams with 5 tags

In [40]:

```
svm(x_train_tfidf,y_tain,x_cv_tfidf,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.4048952499481435
The f1 score of cv data for alpa  0.001 is 0.4103019538188277
The f1 score of cv data for alpa  0.01 is 0.21150033046926633
The f1 score of cv data for alpa  0.1 is 0.30599444223898375
The f1 score of cv data for alpa  1 is 0.28061852047091895
The f1 score of cv data for alpa  10 is 0.18560953253895507
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.29605866177818513
```



we can see from above plot we can say alpha=0.001 we are getting better results

In [50]:

```
best_svm_linear(x_train_tfidf,y_tain,x_test_tfidf,y_test,0.001)
```

```
For values of c =  0.001 The train f1  score is: 0.6066977840213894
For values of  c =  0.001 The test  f1 score is: 0.3989233602493271
Hamming loass on test data is  0.2875635377838021
Precision recall report :
              precision    recall  f1-score   support

           0       0.22      0.43      0.30       320
           1       0.27      0.39      0.32       522
           2       0.50      0.59      0.54       871
           3       0.35      0.66      0.45       573
           4       0.18      0.58      0.27       305

   micro avg       0.32      0.54      0.40      2591
   macro avg       0.30      0.53      0.38      2591
```

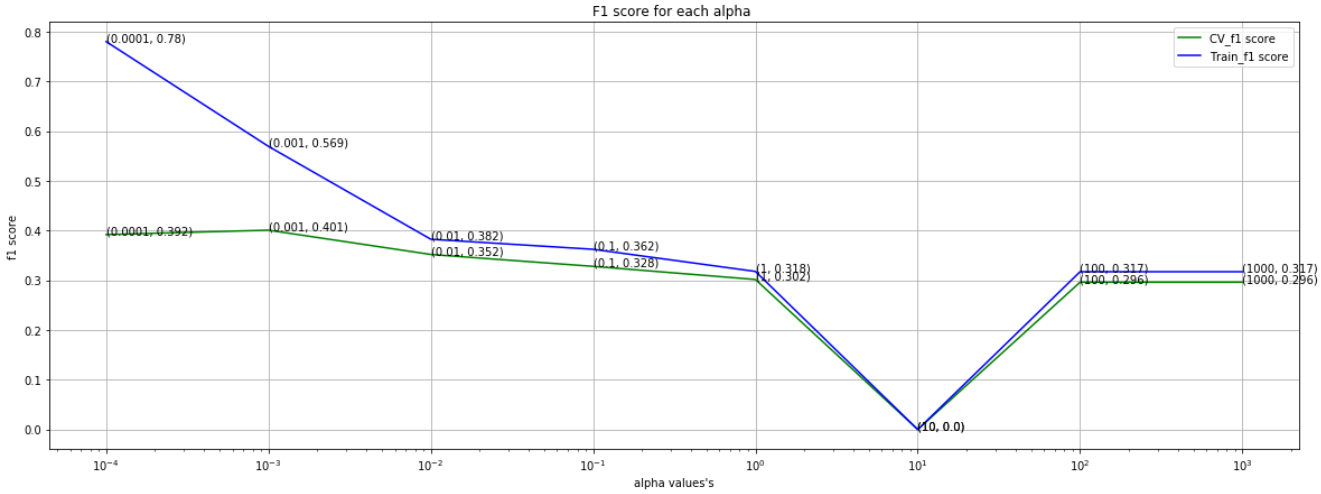
```
weighted avg       0.35      0.54      0.42      2591
 samples avg       0.19      0.27      0.20      2591
```

4.4.3 Unigrams with 5 tags

In [56]:

```
svm(x_train_tfidf1,y_tain,x_cv_tfidf1,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.39183987682832944
The f1 score of cv data for alpa  0.001 is 0.40067283431455003
The f1 score of cv data for alpa  0.01 is 0.3518495910145281
The f1 score of cv data for alpa  0.1 is 0.3279550615513326
The f1 score of cv data for alpa  1 is 0.3015637284878134
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.29605866177818513
The f1 score of cv data for alpa  1000 is 0.29605866177818513
```



we can see from above plot we can say alpha=0.001 we are getting better results

In [57]:

```
best_svm_linear(x_train_tfidf1,y_tain,x_test_tfidf1,y_test,0.001)
```

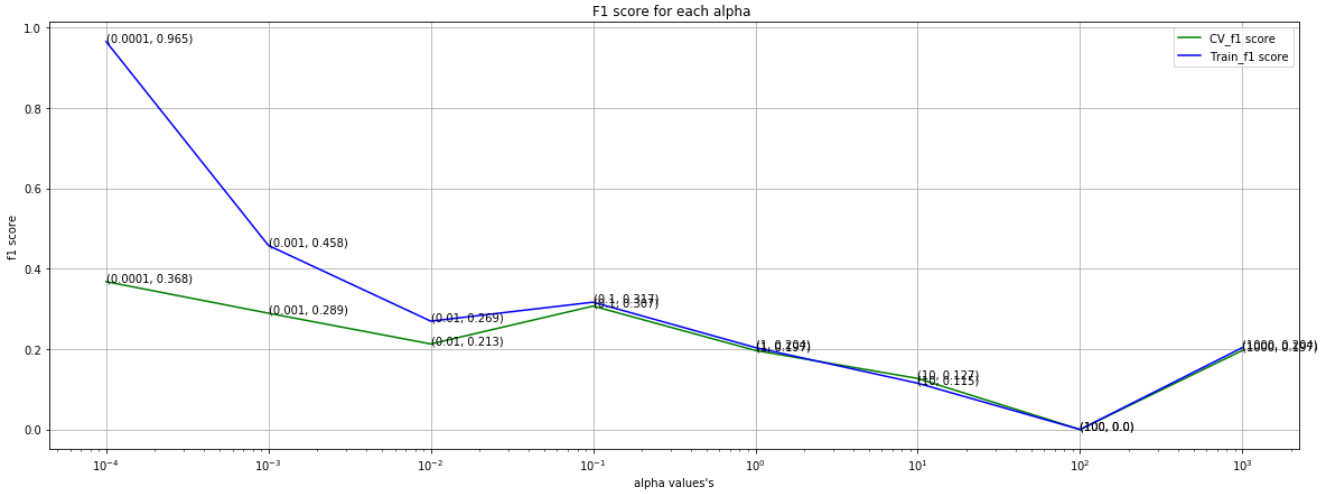
```
For values of c =  0.001 The train f1  score is: 0.5676826153584199
For values of  c =  0.001 The test  f1 score is: 0.3959667529636191
Hamming loass on test data is  0.3004405286343612
Precision recall report :
              precision    recall  f1-score   support

           0       0.21      0.52      0.30       320
           1       0.27      0.42      0.33       522
           2       0.50      0.59      0.54       871
           3       0.34      0.66      0.45       573
           4       0.17      0.58      0.27       305

   micro avg       0.31      0.56      0.40      2591
   macro avg       0.30      0.55      0.38      2591
weighted avg       0.35      0.56      0.42      2591
 samples avg       0.19      0.28      0.21      2591
```

4.4.4 Bigrams with 5 tags

In [52]:

```
svm(x_train_tfidf2,y_tain,x_cv_tfidf2,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.367699642431466
The f1 score of cv data for alpa  0.001 is 0.289229834421996
The f1 score of cv data for alpa  0.01 is 0.2127968098604314
The f1 score of cv data for alpa  0.1 is 0.306979994758452
The f1 score of cv data for alpa  1 is 0.19660861594867093
The f1 score of cv data for alpa  10 is 0.12694775435380384
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.19660861594867093
```



we can see from above plot we can say alpha=0.001 we are getting better results

In [53]:

```
best_svm_linear(x_train_tfidf2,y_tain,x_test_tfidf2,y_test,0.001)
```

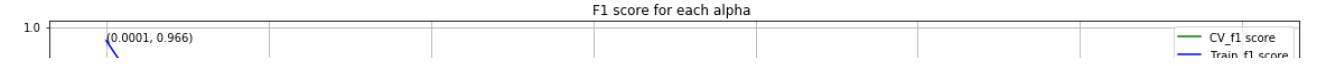
```
For values of c =  0.001 The train f1  score is: 0.6254899722190509
For values of  c =  0.001 The test  f1 score is: 0.38322694254897643
Hamming loass on test data is  0.37980345645543884
Precision recall report :
              precision    recall  f1-score   support

           0       0.25      0.17      0.20       320
           1       0.20      0.90      0.33       522
           2       0.44      0.75      0.56       871
           3       0.23      0.95      0.37       573
           4       0.19      0.05      0.07       305

   micro avg       0.27      0.67      0.38      2591
   macro avg       0.26      0.56      0.31      2591
weighted avg       0.29      0.67      0.37      2591
 samples avg       0.23      0.34      0.25      2591
```

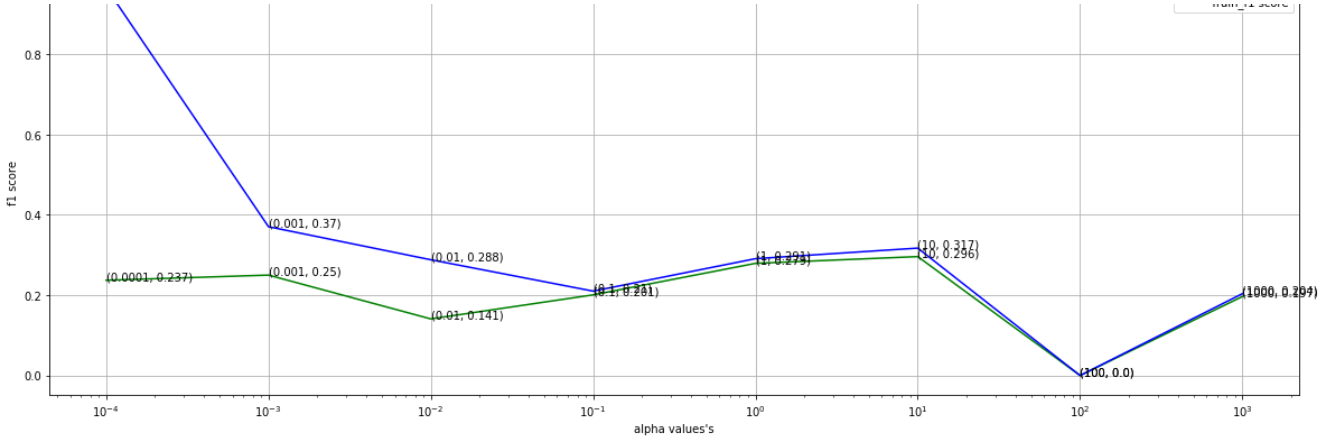
4.4.5 Trigrams with 5 tags

In [58]:

```
svm(x_train_tfidf3,y_tain,x_cv_tfidf3,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.23714652956298196
The f1 score of cv data for alpa  0.001 is 0.25
The f1 score of cv data for alpa  0.01 is 0.14092884923358498
The f1 score of cv data for alpa  0.1 is 0.2007434944237918
The f1 score of cv data for alpa  1 is 0.27919979033807985
The f1 score of cv data for alpa  10 is 0.29605866177818513
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.19660861594867093
```





we can see from above plot we can say alpha=10 we are getting better results

In [59]:

```
best_svm_linear(x_train_tfidf3,y_tain,x_test_tfidf3,y_test,10)
```

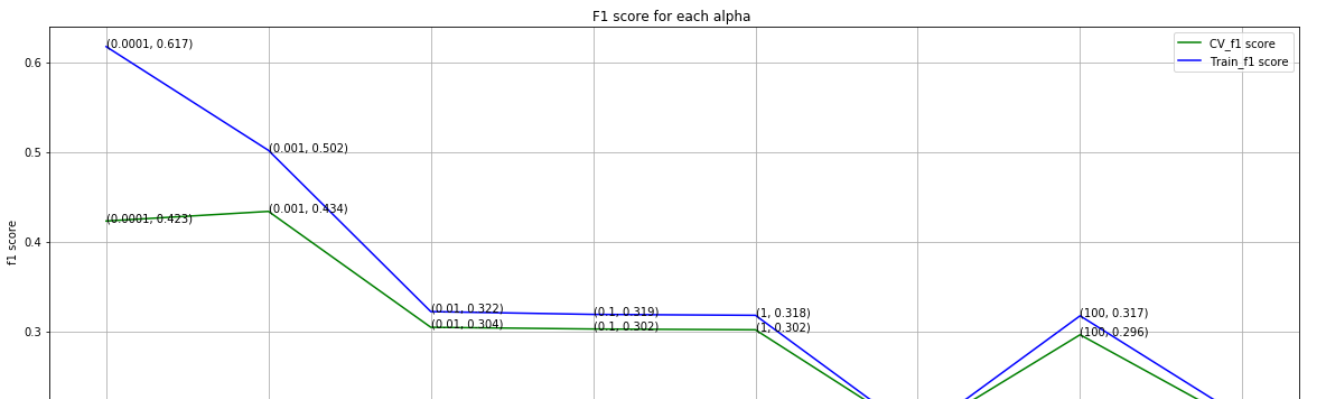
```
For values of c =  10 The train f1  score is: 0.29938094848509644
For values of  c =  10 The test  f1 score is: 0.2967493883257602
Hamming loass on test data is  0.5454422229752626
Precision recall report :
               precision    recall  f1-score   support

            0       0.00      0.00      0.00       320
            1       0.18      1.00      0.30       522
            2       0.30      1.00      0.46       871
            3       0.00      0.00      0.00       573
            4       0.10      1.00      0.19       305

   micro avg       0.19      0.66      0.30      2591
   macro avg       0.12      0.60      0.19      2591
weighted avg       0.15      0.66      0.24      2591
 samples avg       0.19      0.36      0.24      2591
```

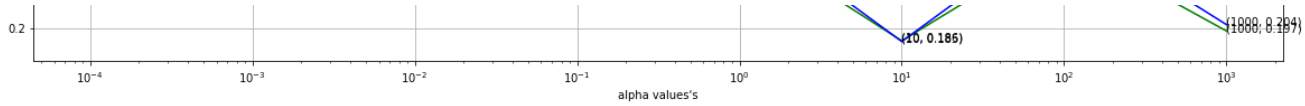
4.4.6 Char-3 grams with 5 tags

In [60]:

```
svm(x_train_tfidf4,y_tain,x_cv_tfidf4,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.4228136882129277
The f1 score of cv data for alpa  0.001 is 0.4335495549856691
The f1 score of cv data for alpa  0.01 is 0.3043592667309885
The f1 score of cv data for alpa  0.1 is 0.3024415857180362
The f1 score of cv data for alpa  1 is 0.3015637284878134
The f1 score of cv data for alpa  10 is 0.18560953253895507
The f1 score of cv data for alpa  100 is 0.29605866177818513
The f1 score of cv data for alpa  1000 is 0.19660861594867093
```





we can see from above plot we can say alpha=0.001 we are getting better results

In [61]:

```
best_svm_linear(x_train_tfidf4,y_tain,x_test_tfidf4,y_test,0.001)
```

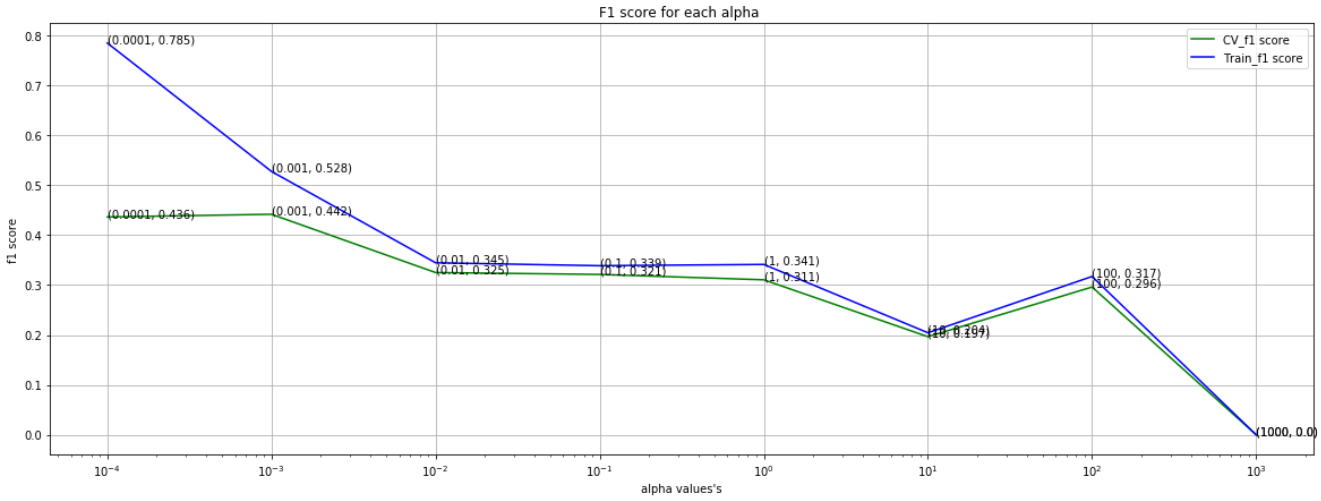
```
For values of c =  0.001 The train f1  score is: 0.49572032214959016
For values of  c =  0.001 The test  f1 score is: 0.4177449168207024
Hamming loass on test data is  0.34157912572009486
Precision recall report :
               precision    recall  f1-score   support

           0       0.19      0.67      0.29       320
           1       0.26      0.65      0.37       522
           2       0.46      0.77      0.57       871
           3       0.35      0.69      0.47       573
           4       0.18      0.61      0.28       305

   micro avg       0.30      0.70      0.42      2591
   macro avg       0.29      0.68      0.40      2591
weighted avg       0.33      0.70      0.44      2591
 samples avg       0.20      0.35      0.23      2591
```

4.4.7 Char-5 grams with 5 tags

In [62]:

```
svm(x_train_tfidf5,y_tain,x_cv_tfidf5,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.43636363636363634
The f1 score of cv data for alpa  0.001 is 0.44173913043478263
The f1 score of cv data for alpa  0.01 is 0.3250318742031449
The f1 score of cv data for alpa  0.1 is 0.3212770256617498
The f1 score of cv data for alpa  1 is 0.3105022831050228
The f1 score of cv data for alpa  10 is 0.19656357388316154
The f1 score of cv data for alpa  100 is 0.29605866177818513
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.001 we are getting better results

In [63]:

```
best_svm_linear(x_train_tfidf5,y_tain,x_test_tfidf5,y_test,0.001)
```

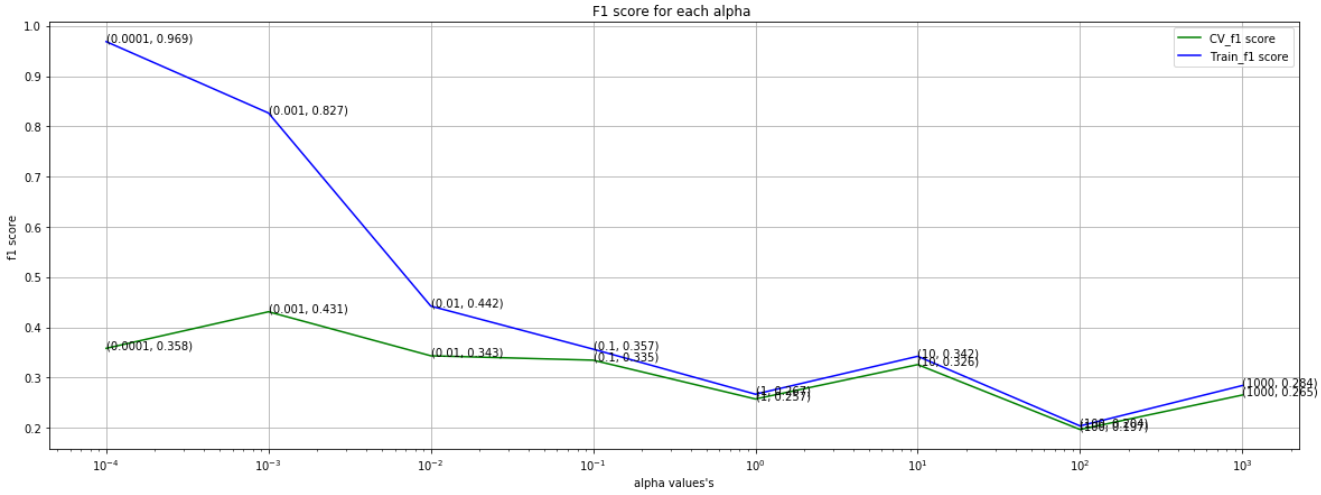
```
For values of c =  0.001 The train f1  score is: 0.5317239736361471
For values of  c =  0.001 The test  f1 score is: 0.42534744358026266
Hamming loass on test data is  0.30545577770247373
Precision recall report :
               precision    recall  f1-score   support

           0       0.20      0.63      0.31       320
           1       0.30      0.53      0.38       522
           2       0.50      0.68      0.58       871
           3       0.35      0.69      0.47       573
           4       0.19      0.64      0.29       305

   micro avg       0.32      0.64      0.43      2591
   macro avg       0.31      0.64      0.41      2591
weighted avg       0.36      0.64      0.45      2591
 samples avg       0.20      0.32      0.22      2591
```

4.4.8 Uni+Bi+Tri grams with 5 tags

In [64]:

```
svm(x_train_tfidf123,y_tain,x_cv_tfidf123,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.3581589958158996
The f1 score of cv data for alpa  0.001 is 0.43098768010022964
The f1 score of cv data for alpa  0.01 is 0.3433075073470478
The f1 score of cv data for alpa  0.1 is 0.33461800045341195
The f1 score of cv data for alpa  1 is 0.2573189522342065
The f1 score of cv data for alpa  10 is 0.32577591899625796
The f1 score of cv data for alpa  100 is 0.19660861594867093
The f1 score of cv data for alpa  1000 is 0.265278810408922
```



By looking above plot we can say alpha=0.01 we are getting better results

In [65]:

```
best_svm_linear(x_train_tfidf123,y_tain,x_test_tfidf123,y_test,0.01)
```

```
For values of c =  0.01 The train f1  score is: 0.4076316181213715
For values of  c =  0.01 The test  f1 score is: 0.3496602938852899
Hamming loass on test data is  0.5579125720094883
Precision recall report :
               precision    recall  f1-score   support

           0       0.26      0.25      0.25       320
           1       0.18      1.00      0.30       522
           2       0.31      0.99      0.47       871
           3       0.19      1.00      0.33       573
           4       0.17      0.57      0.26       305
```

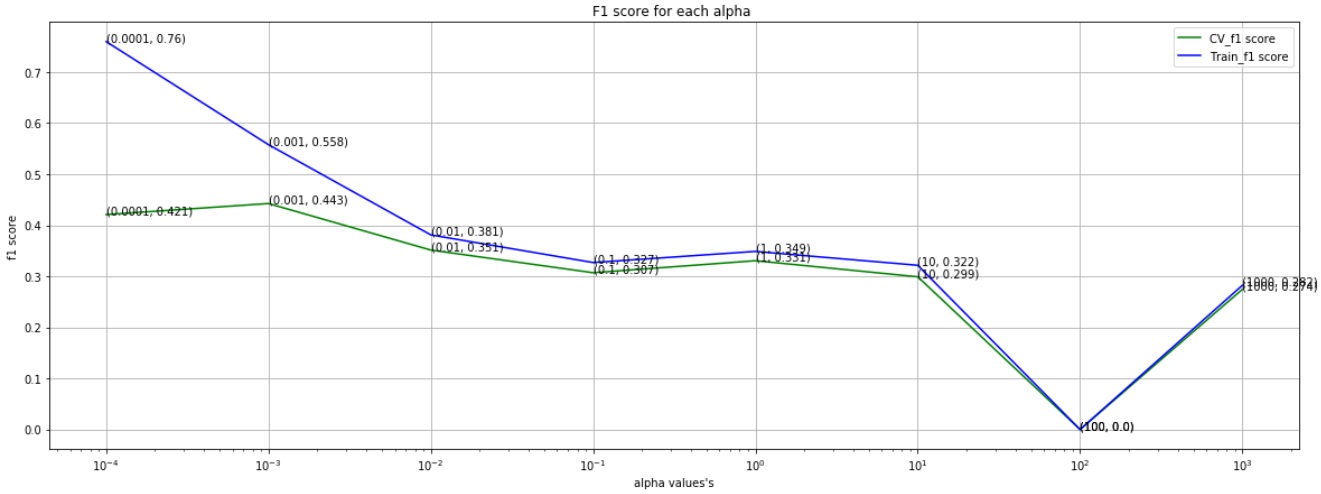
```
   micro avg       0.22      0.85      0.35      2591
   macro avg       0.22      0.76      0.32      2591
weighted avg       0.23      0.85      0.35      2591
 samples avg       0.21      0.44      0.27      2591
```

4.4.9 Char3+Char5 grams with 5 tags

In [66]:

```
svm(xtrain_train_tfidf45,y_tain,xtrain_cv_tfidf45,y_cv)
```

```
The f1 score of cv data for alpa  0.0001 is 0.42091152815013405
The f1 score of cv data for alpa  0.001 is 0.44254278728606355
The f1 score of cv data for alpa  0.01 is 0.3514395963193826
The f1 score of cv data for alpa  0.1 is 0.30704129993229523
The f1 score of cv data for alpa  1 is 0.3305632502308402
The f1 score of cv data for alpa  10 is 0.2990697140231997
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.2744981412639405
```



By looking above plot we can say alpha=0.001 we are getting better results

In [69]:

```
best_svm_linear(xtrain_train_tfidf45,y_tain,xtrain_test_tfidf45,y_test,0.001)
```

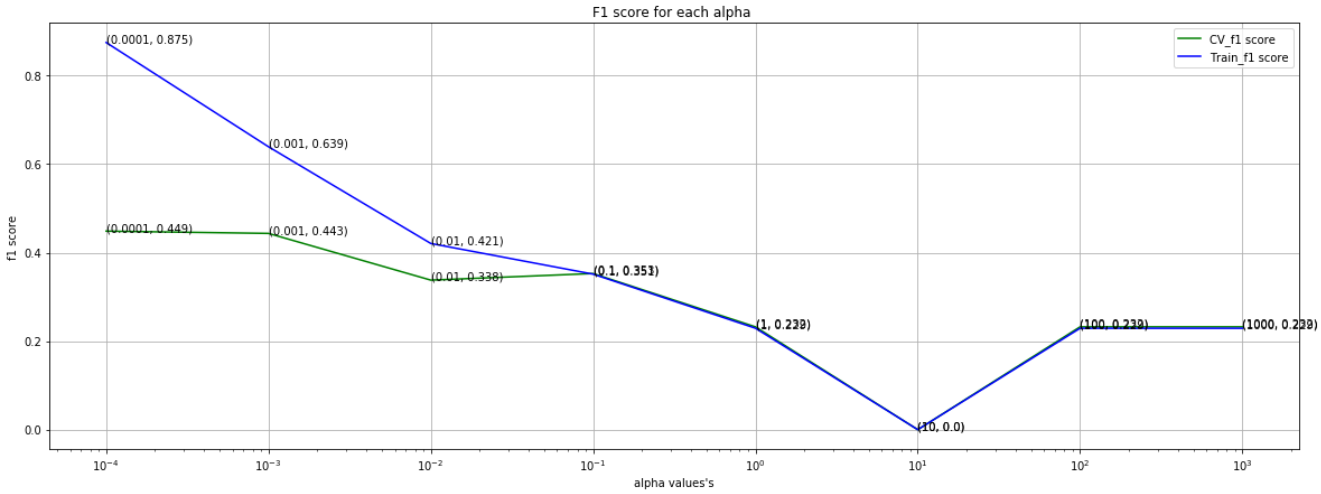
```
For values of c =  0.001 The train f1  score is: 0.6089943785134291
For values of  c =  0.001 The test  f1 score is: 0.4404468301175105
Hamming loass on test data is  0.2614029142663504
Precision recall report :
              precision    recall  f1-score   support

           0       0.24      0.44      0.31       320
           1       0.35      0.38      0.36       522
           2       0.50      0.71      0.58       871
           3       0.37      0.69      0.48       573
           4       0.20      0.54      0.29       305

   micro avg       0.35      0.59      0.44      2591
   macro avg       0.33      0.55      0.41      2591
weighted avg       0.37      0.59      0.45      2591
 samples avg       0.20      0.29      0.22      2591
```

4.4.10 Ngrams with 3 tags

In [70]:

```
svm(x_train_tfidf_th,y_tain1,x_cv_tfidf_th,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.4486187845303867
The f1 score of cv data for alpa  0.001 is 0.4434692823954569
The f1 score of cv data for alpa  0.01 is 0.3378769103676167
The f1 score of cv data for alpa  0.1 is 0.3527914744711309
The f1 score of cv data for alpa  1 is 0.23229750382068262
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.23229750382068262
The f1 score of cv data for alpa  1000 is 0.23229750382068262
```



By looking above plot we can say alpha=0.001 we are getting better results

In [72]:

```
best_svm_linear(x_train_tfidf_th,y_tain1,x_test_tfidf_th,y_test1,0.001)
```

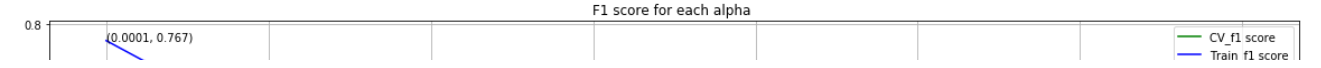
```
For values of c =  0.001 The train f1  score is: 0.6412847847154922
For values of  c =  0.001 The test  f1 score is: 0.45630448649797994
Hamming loass on test data is  0.28882864565683947
Precision recall report :
              precision    recall  f1-score   support

           0       0.26      0.39      0.31       526
           1       0.51      0.57      0.54       868
           2       0.38      0.66      0.48       567

   micro avg       0.39      0.55      0.46      1961
   macro avg       0.38      0.54      0.44      1961
weighted avg       0.40      0.55      0.46      1961
 samples avg       0.19      0.23      0.20      1961
```

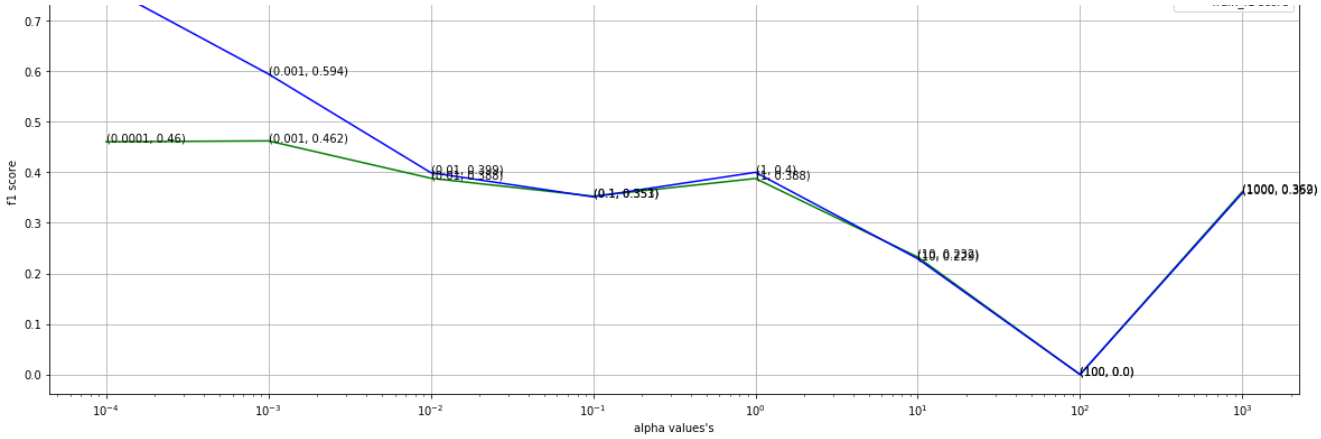
4.4.11 Unigrams with 3 tags

In [73]:

```
svm(x_train_tfidf_th1,y_tain1,x_cv_tfidf_th1,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.46043899948953554
The f1 score of cv data for alpa  0.001 is 0.46216560509554144
The f1 score of cv data for alpa  0.01 is 0.38828295419133685
The f1 score of cv data for alpa  0.1 is 0.3527914744711309
The f1 score of cv data for alpa  1 is 0.3877949263792798
The f1 score of cv data for alpa  10 is 0.23229750382068262
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.36193339500462535
```





By looking above plot we can say alpha=0.001 we are getting better results

In [74]:

```
best_svm_linear(x_train_tfidf_th1,y_tain1,x_test_tfidf_th1,y_test1,0.001)
```

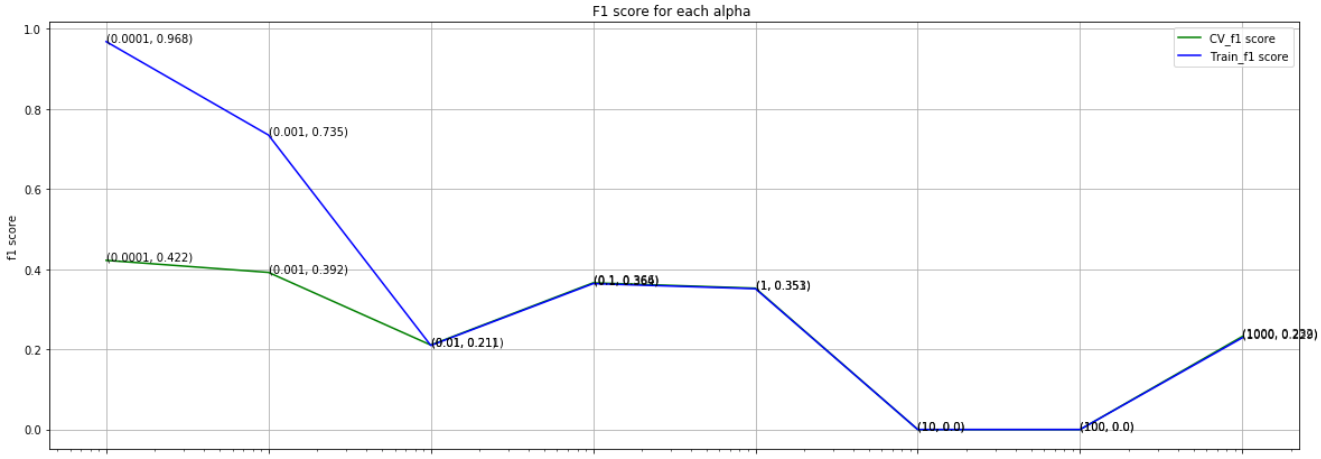
```
For values of c =  0.001 The train f1  score is: 0.5881417120479715
For values of  c =  0.001 The test  f1 score is: 0.4662987780843516
Hamming loass on test data is  0.3058850107308257
Precision recall report :
              precision    recall  f1-score   support

           0       0.24      0.41      0.31       526
           1       0.50      0.64      0.56       868
           2       0.37      0.72      0.49       567

   micro avg       0.38      0.60      0.47      1961
   macro avg       0.37      0.59      0.45      1961
weighted avg       0.39      0.60      0.47      1961
 samples avg       0.21      0.25      0.22      1961
```

4.4.12 BiGrams with 3 tags

In [75]:

```
svm(x_train_tfidf_th2,y_tain1,x_cv_tfidf_th2,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.42231947483588617
The f1 score of cv data for alpa  0.001 is 0.39215686274509803
The f1 score of cv data for alpa  0.01 is 0.2109035065267469
The f1 score of cv data for alpa  0.1 is 0.36615237792269767
The f1 score of cv data for alpa  1 is 0.3527914744711309
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.23229750382068262
```



By looking above plot we can say alpha=1 we are getting better results

In [76]:

```
best_svm_linear(x_train_tfidf_th2,y_tain1,x_test_tfidf_th2,y_test1,1)
```

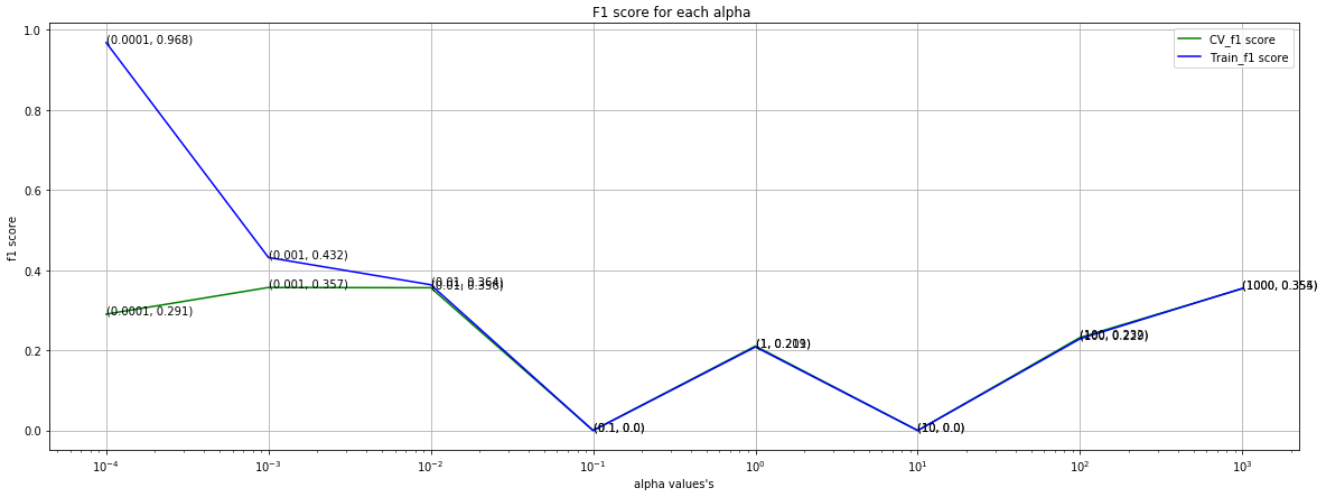
```
For values of c =  1 The train f1  score is: 0.2290720726482062
For values of  c =  1 The test  f1 score is: 0.23086319218241044
Hamming loass on test data is  0.4267479950299334
Precision recall report :
               precision    recall  f1-score   support

           0       0.00      0.00      0.00       526
           1       0.00      0.00      0.00       868
           2       0.19      1.00      0.32       567

   micro avg       0.19      0.29      0.23      1961
   macro avg       0.06      0.33      0.11      1961
weighted avg       0.06      0.29      0.09      1961
 samples avg       0.19      0.12      0.14      1961
```

4.4.13 Trigrams with 3 tags

In [77]:

```
svm(x_train_tfidf_th3,y_tain1,x_cv_tfidf_th3,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.290519877675841
The f1 score of cv data for alpa  0.001 is 0.357106727364694
The f1 score of cv data for alpa  0.01 is 0.3564356435643565
The f1 score of cv data for alpa  0.1 is 0.0
The f1 score of cv data for alpa  1 is 0.2109016811003566
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.23229750382068262
The f1 score of cv data for alpa  1000 is 0.35404992358634746
```



By looking above plot we can say alpha=0.001 we are getting better results

In [78]:

```
best_svm_linear(x_train_tfidf_th3,y_tain1,x_test_tfidf_th3,y_test1,0.001)
```

```
For values of c =  0.001 The train f1  score is: 0.4287995134189918
For values of  c =  0.001 The test  f1 score is: 0.35684542586750795
Hamming loass on test data is  0.5757370382921043
Precision recall report :
```

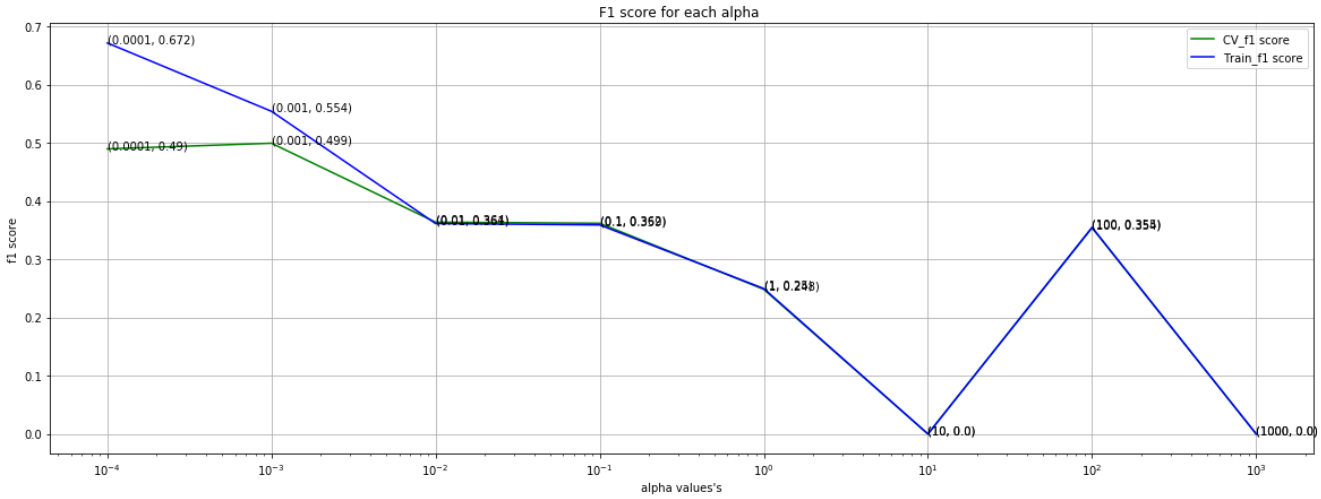
```
                precision    recall  f1-score   support

           0       0.18      1.00      0.30       526
           1       0.29      1.00      0.45       868
           2       0.32      0.04      0.07       567

   micro avg       0.24      0.72      0.36      1961
   macro avg       0.27      0.68      0.27      1961
weighted avg       0.27      0.72      0.30      1961
 samples avg       0.24      0.33      0.26      1961
```

4.4.15 Char3 grams with 3 tags

In [79]:

```
svm(x_train_tfidf_th4,y_tain1,x_cv_tfidf_th4,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.48965056754952146
The f1 score of cv data for alpa  0.001 is 0.49940205692418077
The f1 score of cv data for alpa  0.01 is 0.3637842863784287
The f1 score of cv data for alpa  0.1 is 0.36193339500462535
The f1 score of cv data for alpa  1 is 0.24838869608329203
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.35404992358634746
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.001 we are getting better results

In [80]:

```
best_svm_linear(x_train_tfidf_th4,y_tain1,x_test_tfidf_th4,y_test1,0.001)
```

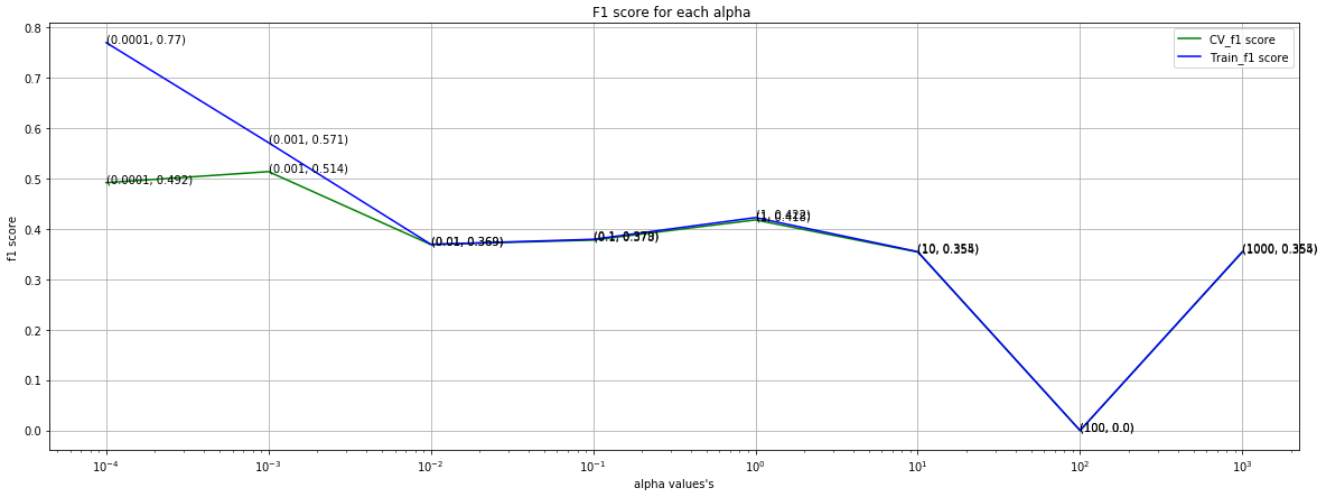
```
For values of c =  0.001 The train f1  score is: 0.5464817668207499
For values of  c =  0.001 The test  f1 score is: 0.5005413208228077
Hamming loass on test data is  0.31266237433638316
Precision recall report :
                precision    recall  f1-score   support

           0       0.30      0.52      0.38       526
           1       0.50      0.76      0.60       868
           2       0.33      0.79      0.47       567

   micro avg       0.39      0.71      0.50      1961
   macro avg       0.38      0.69      0.49      1961
weighted avg       0.40      0.71      0.51      1961
 samples avg       0.22      0.30      0.24      1961
```

4.4.16 char5 grams with 3 tags

In [81]:

```
svm(x_train_tfidf_th5,y_tain1,x_cv_tfidf_th5,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.49178695868591343
The f1 score of cv data for alpa  0.001 is 0.5137236962488564
The f1 score of cv data for alpa  0.01 is 0.3690377559474494
The f1 score of cv data for alpa  0.1 is 0.37785095743383335
The f1 score of cv data for alpa  1 is 0.417923125929072
The f1 score of cv data for alpa  10 is 0.35404992358634746
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.35404992358634746
```



By looking above plot we can say alpha=0.001 we are getting better results

In [82]:

```
best_svm_linear(x_train_tfidf_th5,y_tain1,x_test_tfidf_th5,y_test1,0.001)
```

```
For values of c =  0.001 The train f1  score is: 0.5776082941071322
For values of  c =  0.001 The test  f1 score is: 0.5079666160849773
Hamming loass on test data is  0.2930080198802666
Precision recall report :
              precision    recall  f1-score   support

           0       0.29      0.55      0.38       526
           1       0.53      0.72      0.61       868
           2       0.37      0.75      0.50       567

   micro avg       0.40      0.68      0.51      1961
   macro avg       0.40      0.67      0.50      1961
weighted avg       0.42      0.68      0.52      1961
 samples avg       0.22      0.28      0.23      1961
```

In [ ]:

```

```

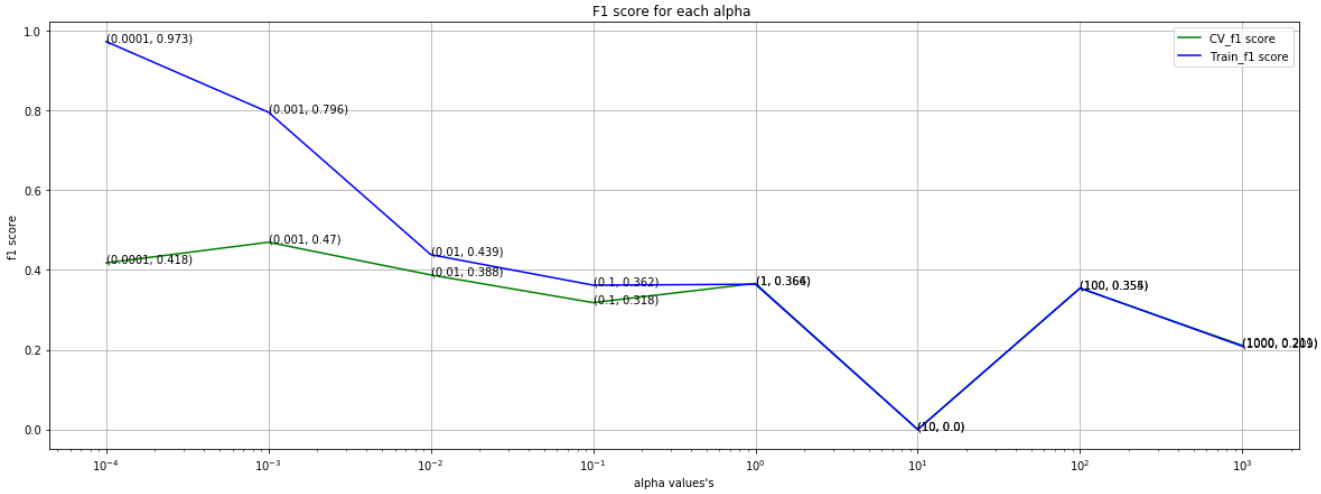
4.4.17 Uni+Bi+Tri grams with 3 tags

In [83]:

```
svm(x_train_tfidf_th123,y_tain1,x_cv_tfidf_th123,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.41791044776119407
The f1 score of cv data for alpa  0.001 is 0.469737546866631
```

```
The f1 score of cv data for alpa  0.01 is 0.3676781910535802
The f1 score of cv data for alpa  0.1 is 0.3182401466544455
The f1 score of cv data for alpa  1 is 0.36641221374045807
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.35404992358634746
The f1 score of cv data for alpa  1000 is 0.2109016811003566
```



By looking above plot we can say alpha=0.001 we are getting better results

In [84]:

```
best_svm_linear(x_train_tfidf_th123,y_tain1,x_test_tfidf_th123,y_test1,0.001)
```

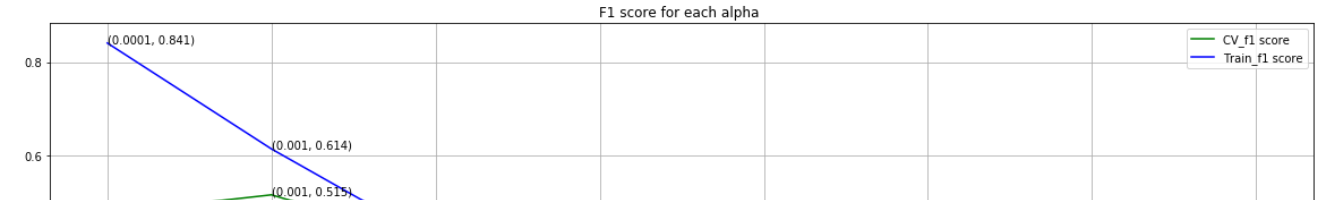
```
For values of c =  0.001 The train f1  score is: 0.7993492407809112
For values of  c =  0.001 The test  f1 score is: 0.4691840277777778
Hamming loass on test data is  0.2762905229865582
Precision recall report :
               precision    recall  f1-score   support

           0       0.27      0.34      0.30       526
           1       0.52      0.62      0.56       868
           2       0.39      0.65      0.49       567

   micro avg       0.41      0.55      0.47      1961
   macro avg       0.39      0.53      0.45      1961
weighted avg       0.41      0.55      0.47      1961
 samples avg       0.20      0.23      0.20      1961
```

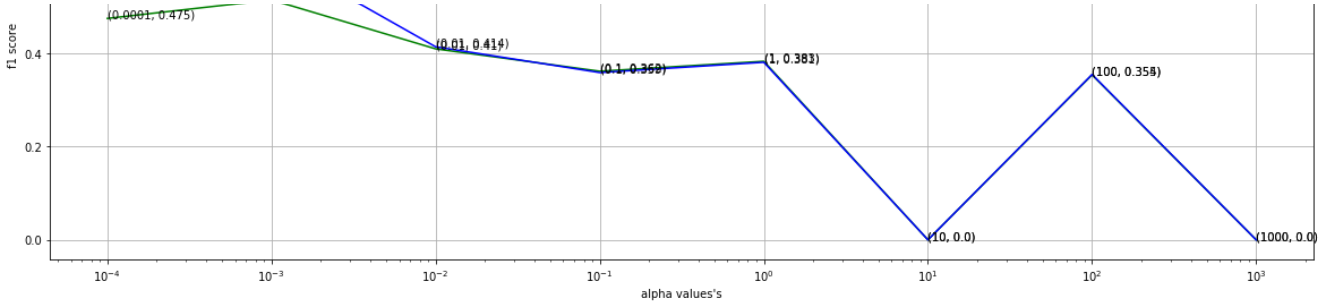
4.4.18 Char3+char5 grams with 3tags

In [85]:

```
svm(x_train_tfidf_th45,y_tain1,x_cv_tfidf_th45,y_cv1)
```

```
The f1 score of cv data for alpa  0.0001 is 0.4752475247524752
The f1 score of cv data for alpa  0.001 is 0.5151725735464442
The f1 score of cv data for alpa  0.01 is 0.4099738255958121
The f1 score of cv data for alpa  0.1 is 0.36193339500462535
The f1 score of cv data for alpa  1 is 0.38325604712369965
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.35404992358634746
The f1 score of cv data for alpa  1000 is 0.0
```





By looking above plot we can say alpha=0.001 we are getting better results

In [86]:

```
best_svm_linear(x_train_tfidf_th45,y_tain1,x_test_tfidf_th45,y_test1,0.001)
```

```
For values of c =  0.001 The train f1  score is: 0.614574898785425
For values of  c =  0.001 The test  f1 score is: 0.5155131264916467
Hamming loass on test data is  0.29809104258443464
Precision recall report :
               precision    recall  f1-score   support

           0       0.28      0.60      0.38       526
           1       0.50      0.80      0.61       868
           2       0.41      0.69      0.52       567

   micro avg       0.40      0.72      0.52      1961
   macro avg       0.40      0.70      0.50      1961
weighted avg       0.41      0.72      0.52      1961
 samples avg       0.23      0.30      0.25      1961
```

4.5 Lositic Regression with OnevsRest

Defining a function to train logistic regression model

In [87]:

```
import seaborn as sns
def lr(x_train,y_train,x_cv,y_cv):

    alpha = [10**x for x in range(-8,4)]

    cv_log_error =[]
    train_log_error=[]
    for i in alpha:

        clf_dt = OneVsRestClassifier(SGDClassifier(loss='log',alpha=i,penalty='l2',class_weight='ba
lanced'))

        clf_dt.fit(x_train,y_train)
        y_pred_prob = clf_dt.predict_proba(x_train)
        t = 0.5 # threshold value
        predict_ytrain = (y_pred_prob >= t).astype(int)

        predict_y_cv = clf_dt.predict_proba(x_cv)

        predict_y = (predict_y_cv >= t).astype(int)
        f1score_train = f1_score(y_train,predict_ytrain,average='micro')
        f1score_cv = f1_score(y_cv,predict_y,average='micro')
        train_log_error.append(f1score_train)
        print('The f1 score of cv data for alpa ',i,'is',f1score_cv)
        cv_log_error.append(f1score_cv)
    fig, ax = plt.subplots(figsize=(20,7))
    plt.xscale('log')
    ax.plot(alpha, cv_log_error,c='g',label='CV_f1 score')
    for i, txt in enumerate(np.round(cv_log_error,3)):
        ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],cv_log_error[i]))
    ax.plot(alpha, train_log_error,c='b',label='Train_f1 score')
```

```
    for i, txt in enumerate(np.round(train_log_error,3)):
        ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],train_log_error[i]))

    plt.grid()
    plt.title("F1 score for each alpha ")
    plt.xlabel("alpha values's",)
    plt.ylabel("f1 score")

    plt.legend()
    plt.show()
```

model with best hyperparameter

In [88]:

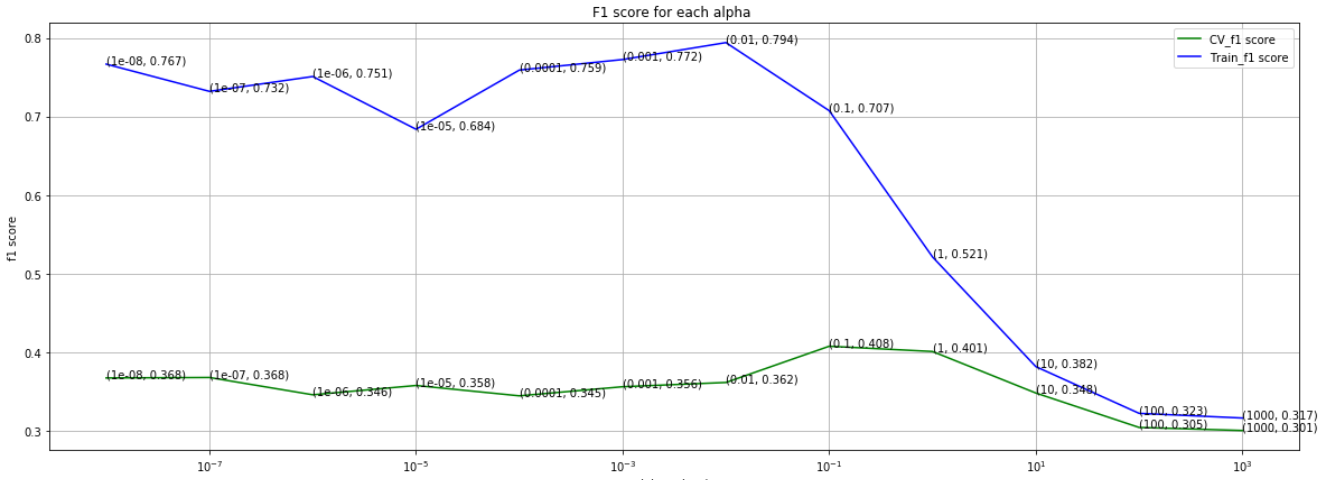
```
def best_lr(x_train,y_train,x_test,y_test,c):



        clf_dt_best = OneVsRestClassifier(SGDClassifier(alpha=c,loss='log',penalty='l2',class_weigh
t='balanced'))
        clf_dt_best.fit(x_train, y_train)
        predict_y_train = clf_dt_best.predict(x_train)
        print('For values of c = ', c, "The train f1  score is:",f1_score(y_train,predict_y_train,a
verage='micro'))
        predict_y_test = clf_dt_best.predict(x_test)
        print('For values of  c = ', c, "The test  f1 score is:",f1_score(y_test,predict_y_test,ave
rage='micro'))

        acc = hamming_loss(y_test,predict_y_test)
        print('Hamming loass on test data is ',acc)


        print("Precision recall report :\n",classification_report(y_test, predict_y_test))
```

4.5.1 Bag of words with 5 tags

In [89]:

```
lr(x_train_bag1,y_tain,x_cv_bag1,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.3675033531327841
The f1 score of cv data for alpa  1e-07 is 0.3681048607318405
The f1 score of cv data for alpa  1e-06 is 0.3460885956644675
The f1 score of cv data for alpa  1e-05 is 0.3579193845124603
The f1 score of cv data for alpa  0.0001 is 0.34476042072458124
The f1 score of cv data for alpa  0.001 is 0.3563352826510721
The f1 score of cv data for alpa  0.01 is 0.3617431546471269
The f1 score of cv data for alpa  0.1 is 0.40783744557329465
The f1 score of cv data for alpa  1 is 0.4011519078473722
The f1 score of cv data for alpa  10 is 0.34834955352663793
The f1 score of cv data for alpa  100 is 0.30456676506337904
The f1 score of cv data for alpa  1000 is 0.30066511739722734
```



By looking above plot we can say alpha=1 we are getting better results

In [91]:

```
best_lr(x_train_bag1,y_tain,x_test_bag1,y_test,1)
```

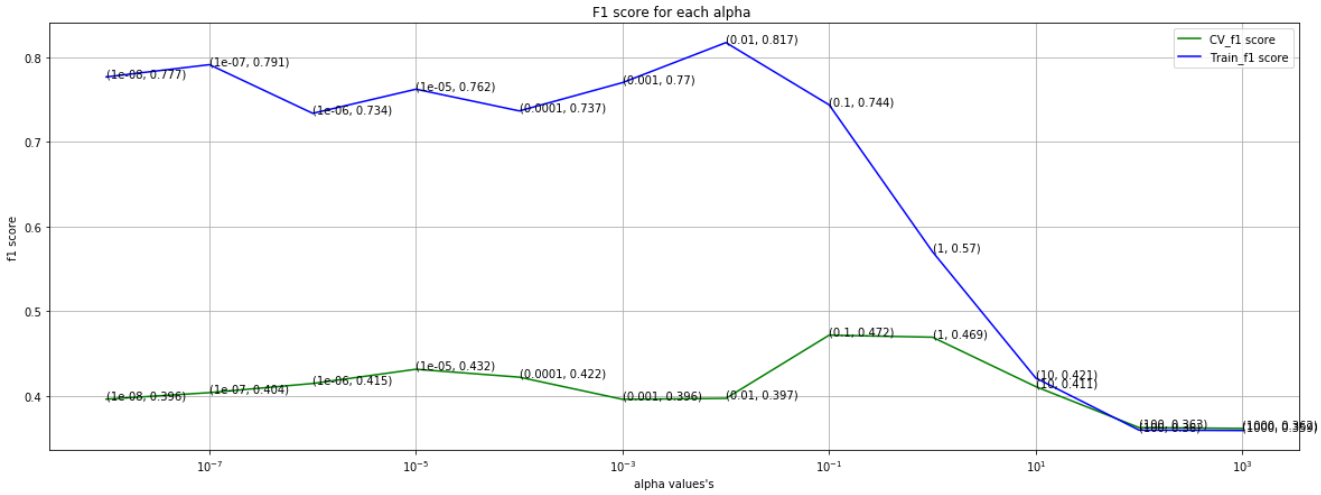
```
For values of c =  1 The train f1  score is: 0.5334041101084401
For values of  c =  1 The test  f1 score is: 0.4061895551257253
Hamming loass on test data is  0.33290410030498135
Precision recall report :
               precision    recall  f1-score   support

           0       0.17      0.70      0.28       320
           1       0.25      0.50      0.34       522
           2       0.48      0.72      0.57       871
           3       0.32      0.74      0.45       573
           4       0.19      0.46      0.27       305

   micro avg       0.30      0.65      0.41      2591
   macro avg       0.28      0.63      0.38      2591
weighted avg       0.33      0.65      0.43      2591
 samples avg       0.20      0.32      0.23      2591
```

4.5.2 Bag of Words with 3 tags

In [93]:

```
lr(x_train_bag_1,y_tain1,x_cv_bag_1,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.3963723659642571
The f1 score of cv data for alpa  1e-07 is 0.40402969247083775
The f1 score of cv data for alpa  1e-06 is 0.41485335856196787
The f1 score of cv data for alpa  1e-05 is 0.43166175024582104
The f1 score of cv data for alpa  0.0001 is 0.4223144306131858
The f1 score of cv data for alpa  0.001 is 0.3961450672077098
The f1 score of cv data for alpa  0.01 is 0.397275204359673
The f1 score of cv data for alpa  0.1 is 0.4719446579554189
The f1 score of cv data for alpa  1 is 0.46947549441100606
The f1 score of cv data for alpa  10 is 0.4111379457309277
The f1 score of cv data for alpa  100 is 0.36279285548596607
The f1 score of cv data for alpa  1000 is 0.36180555555555555
```



By looking above plot we can say alpha=1 we are getting better results

In [95]:

```
best_lr(x_train_bag_1,y_tain1,x_test_bag_1,y_test1,1)
```

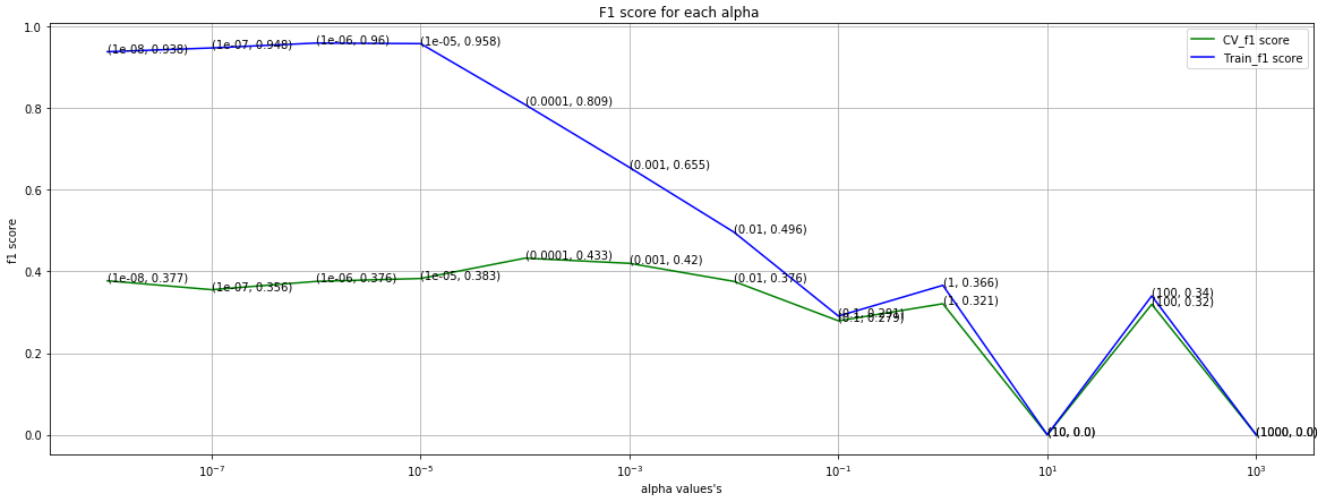
```
For values of c =  1 The train f1  score is: 0.5571466037241019
For values of  c =  1 The test  f1 score is: 0.4660066006600661
Hamming loass on test data is  0.36552581045973115
Precision recall report :
              precision    recall  f1-score   support

           0       0.22      0.66      0.34       526
           1       0.48      0.74      0.59       868
           2       0.34      0.74      0.47       567

   micro avg       0.34      0.72      0.47      1961
   macro avg       0.35      0.71      0.46      1961
weighted avg       0.37      0.72      0.49      1961
 samples avg       0.22      0.31      0.25      1961
```

4.5.3 Ngrams with 5 tags

In [96]:

```
lr(x_train_tfidf,y_tain,x_cv_tfidf,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.37736585365853653
The f1 score of cv data for alpa  1e-07 is 0.35552654482158397
The f1 score of cv data for alpa  1e-06 is 0.37582746102925474
The f1 score of cv data for alpa  1e-05 is 0.38291746641074864
The f1 score of cv data for alpa  0.0001 is 0.4327146171693736
The f1 score of cv data for alpa  0.001 is 0.4200283085633404
The f1 score of cv data for alpa  0.01 is 0.3758070892080643
The f1 score of cv data for alpa  0.1 is 0.27919979033807985
The f1 score of cv data for alpa  1 is 0.32110424990919
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.31970260223048325
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.01 we are getting better results

In [99]:

```
best_lr(x_train_tfidf,y_tain,x_test_tfidf,y_test,0.01)
```

```
For values of c =  0.01 The train f1  score is: 0.48863871814351817
For values of  c =  0.01 The test  f1 score is: 0.3889009793253536
Hamming loass on test data is  0.38061674008810575
Precision recall report :
              precision    recall  f1-score   support

           0       0.21      0.46      0.29       320
           1       0.27      0.29      0.28       522
           2       0.30      1.00      0.46       871
```

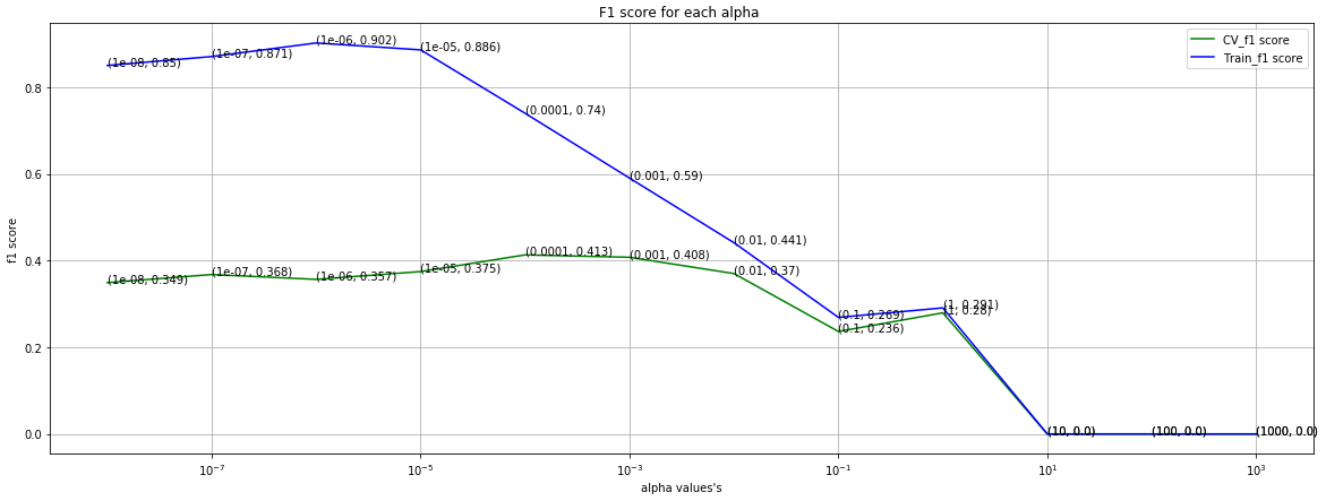
```
           3       0.27      0.86      0.41       573
           4       0.21      0.41      0.28       305

   micro avg       0.27      0.69      0.39      2591
   macro avg       0.25      0.60      0.34      2591
weighted avg       0.27      0.69      0.37      2591
 samples avg       0.24      0.35      0.26      2591
```

4.5.4 Unigrms with 5 tags

In [100]:

```
lr(x_train_tfidf1,y_tain,x_cv_tfidf1,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.34920007900454275
The f1 score of cv data for alpa  1e-07 is 0.36789570177225506
The f1 score of cv data for alpa  1e-06 is 0.3567658855815917
The f1 score of cv data for alpa  1e-05 is 0.3747688514485309
The f1 score of cv data for alpa  0.0001 is 0.4132605010359766
The f1 score of cv data for alpa  0.001 is 0.4077475371514443
The f1 score of cv data for alpa  0.01 is 0.3701033405431387
The f1 score of cv data for alpa  0.1 is 0.236495565708143
The f1 score of cv data for alpa  1 is 0.2795661301609517
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.001 we are getting better results

In [101]:

```
best_lr(x_train_tfidf1,y_tain,x_test_tfidf1,y_test,0.001)
```

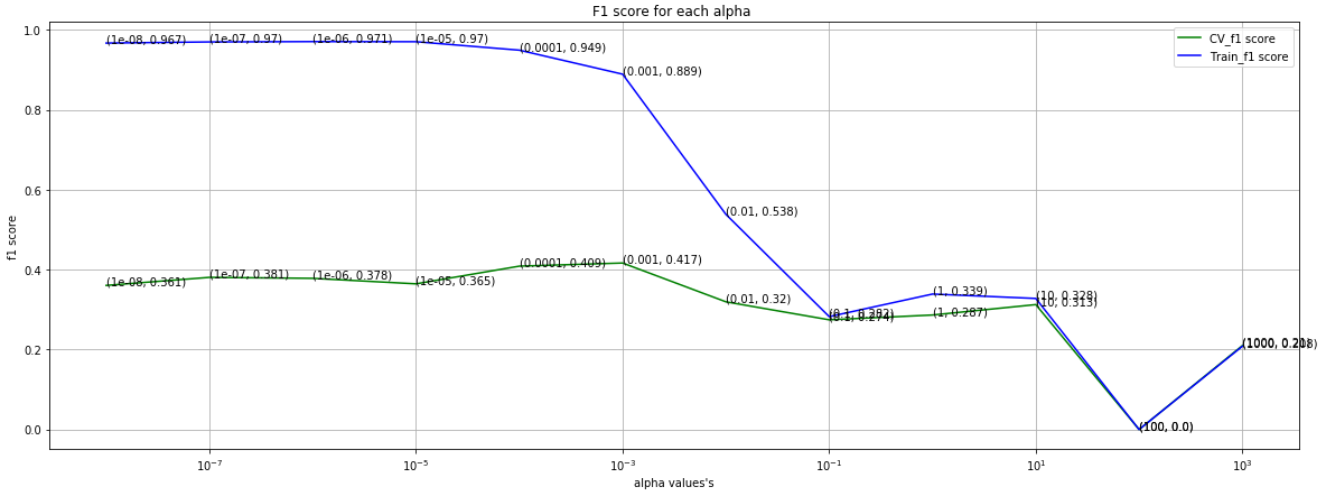
```
For values of c =  0.001 The train f1  score is: 0.5864649151978716
For values of  c =  0.001 The test  f1 score is: 0.4060110730292644
Hamming loass on test data is  0.30538800406641814
Precision recall report :
              precision    recall  f1-score   support

           0       0.20      0.53      0.29       320
           1       0.25      0.50      0.34       522
           2       0.49      0.65      0.56       871
           3       0.34      0.65      0.45       573
           4       0.19      0.55      0.28       305

   micro avg       0.31      0.59      0.41      2591
   macro avg       0.30      0.58      0.38      2591
weighted avg       0.34      0.59      0.42      2591
 samples avg       0.20      0.30      0.22      2591
```

4.5.5 Bigrams with 5 tags

In [102]:

```
lr(x_train_tfidf2,y_tain,x_cv_tfidf2,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.36055747040311703
The f1 score of cv data for alpa  1e-07 is 0.38071383844708834
The f1 score of cv data for alpa  1e-06 is 0.3782642089093702
The f1 score of cv data for alpa  1e-05 is 0.3645141822570911
The f1 score of cv data for alpa  0.0001 is 0.408770555990603
The f1 score of cv data for alpa  0.001 is 0.41660867548132685
The f1 score of cv data for alpa  0.01 is 0.31952247191011235
The f1 score of cv data for alpa  0.1 is 0.2744981412639405
The f1 score of cv data for alpa  1 is 0.2865021770682148
The f1 score of cv data for alpa  10 is 0.31256505576208177
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.20996282527881038
```



By looking above plot we can say alpha=0.01 we are getting better results

In [105]:

```
best_lr(x_train_tfidf2,y_tain,x_test_tfidf2,y_test,0.01)
```

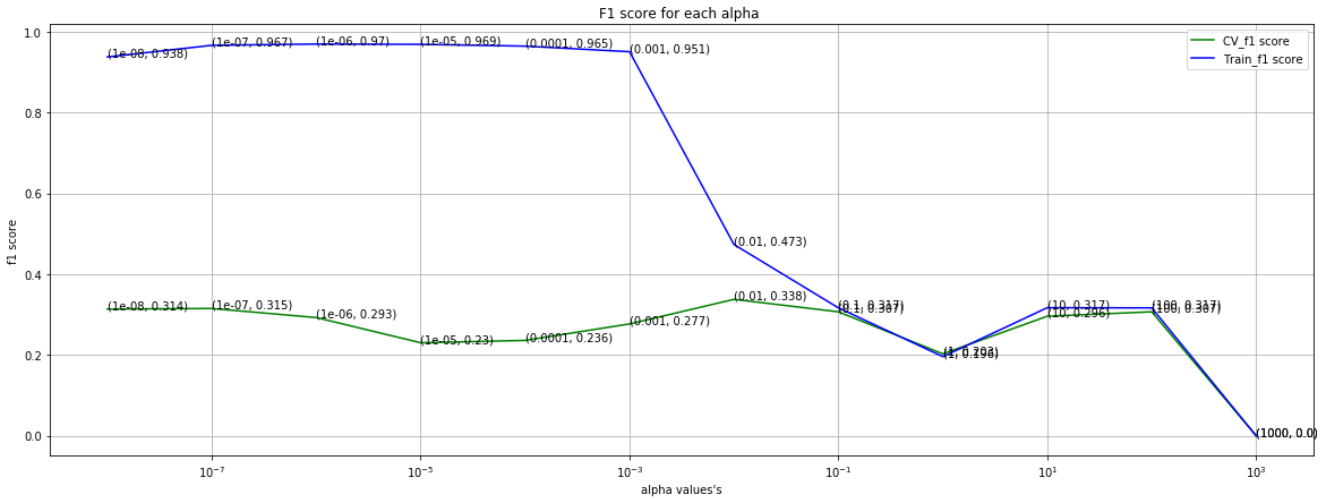
```
For values of c =  0.01 The train f1  score is: 0.647900182592818
For values of  c =  0.01 The test  f1 score is: 0.3446226975638741
Hamming loass on test data is  0.29901728227719415
Precision recall report :
              precision    recall  f1-score   support

           0       0.15      0.85      0.25       320
           1       0.28      0.25      0.27       522
           2       0.41      0.82      0.55       871
           3       0.64      0.06      0.12       573
           4       0.26      0.02      0.04       305

   micro avg       0.28      0.45      0.34      2591
   macro avg       0.35      0.40      0.24      2591
weighted avg       0.38      0.45      0.30      2591
 samples avg       0.19      0.22      0.19      2591
```

4.5.6 Trigrams with 5 tags

In [106]:

```
lr(x_train_tfidf3,y_tain,x_cv_tfidf3,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.31381733021077285
The f1 score of cv data for alpa  1e-07 is 0.31514368743945753
The f1 score of cv data for alpa  1e-06 is 0.29275484414490316
The f1 score of cv data for alpa  1e-05 is 0.23039690222652467
The f1 score of cv data for alpa  0.0001 is 0.23617619493908154
The f1 score of cv data for alpa  0.001 is 0.2769926651215376
The f1 score of cv data for alpa  0.01 is 0.33782771535580525
The f1 score of cv data for alpa  0.1 is 0.306979994758452
The f1 score of cv data for alpa  1 is 0.20288561654023501
The f1 score of cv data for alpa  10 is 0.29605866177818513
The f1 score of cv data for alpa  100 is 0.306979994758452
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.01 we are getting better results

In [110]:

```
best_lr(x_train_tfidf3,y_tain,x_test_tfidf3,y_test,0.01)
```

```
For values of c =  0.01 The train f1  score is: 0.46890194204772206
For values of  c =  0.01 The test  f1 score is: 0.3244281182167307
Hamming loass on test data is  0.5484242629617079
Precision recall report :
              precision    recall  f1-score   support

           0       0.15      0.47      0.23       320
           1       0.18      1.00      0.30       522
           2       0.30      1.00      0.46       871
           3       0.32      0.29      0.30       573
           4       0.12      0.78      0.21       305

   micro avg       0.21      0.75      0.32      2591
   macro avg       0.21      0.71      0.30      2591
weighted avg       0.24      0.75      0.33      2591
 samples avg       0.20      0.40      0.25      2591
```

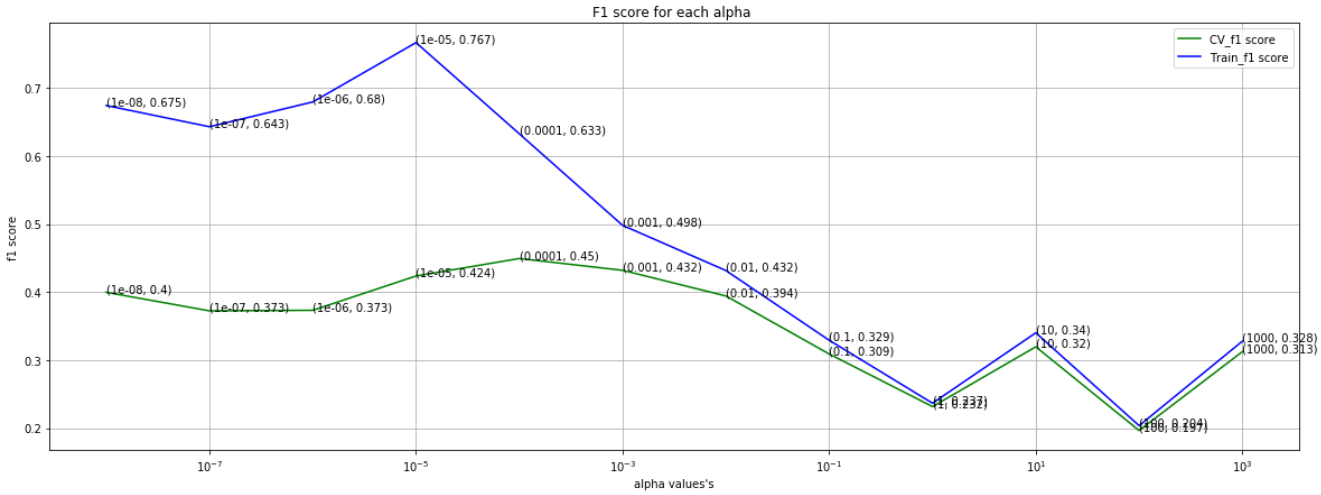
4.5.7 Char3 grams with 5 tags

In [111]:

```
lr(x_train_tfidf4,y_tain,x_cv_tfidf4,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.3998646820027063
The f1 score of cv data for alpa  1e-07 is 0.3725888324873096
The f1 score of cv data for alpa  1e-06 is 0.37337845459672864
The f1 score of cv data for alpa  1e-05 is 0.42382115348487687
The f1 score of cv data for alpa  0.0001 is 0.4495587471880948
The f1 score of cv data for alpa  0.001 is 0.43218980442449506
The f1 score of cv data for alpa  0.01 is 0.394445809781273
```

```
The f1 score of cv data for alpa  0.1 is 0.3091195518991773
The f1 score of cv data for alpa  1 is 0.2317691121392377
The f1 score of cv data for alpa  10 is 0.31970260223048325
The f1 score of cv data for alpa  100 is 0.19660861594867093
The f1 score of cv data for alpa  1000 is 0.31256505576208177
```



By looking above plot we can say alpha=0.0001 we are getting better results

In [114]:

```
best_lr(x_train_tfidf4,y_tain,x_test_tfidf4,y_test,0.0001)
```

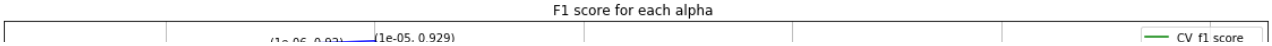
```
For values of c =  0.0001 The train f1  score is: 0.6145524481128296
For values of  c =  0.0001 The test  f1 score is: 0.4295701708959089
Hamming loass on test data is  0.2986106404608607
Precision recall report :
              precision    recall  f1-score   support

           0       0.21      0.50      0.29       320
           1       0.30      0.52      0.38       522
           2       0.47      0.73      0.57       871
           3       0.36      0.73      0.48       573
           4       0.19      0.59      0.29       305

   micro avg       0.32      0.64      0.43      2591
   macro avg       0.30      0.61      0.40      2591
weighted avg       0.35      0.64      0.44      2591
 samples avg       0.21      0.32      0.23      2591
```

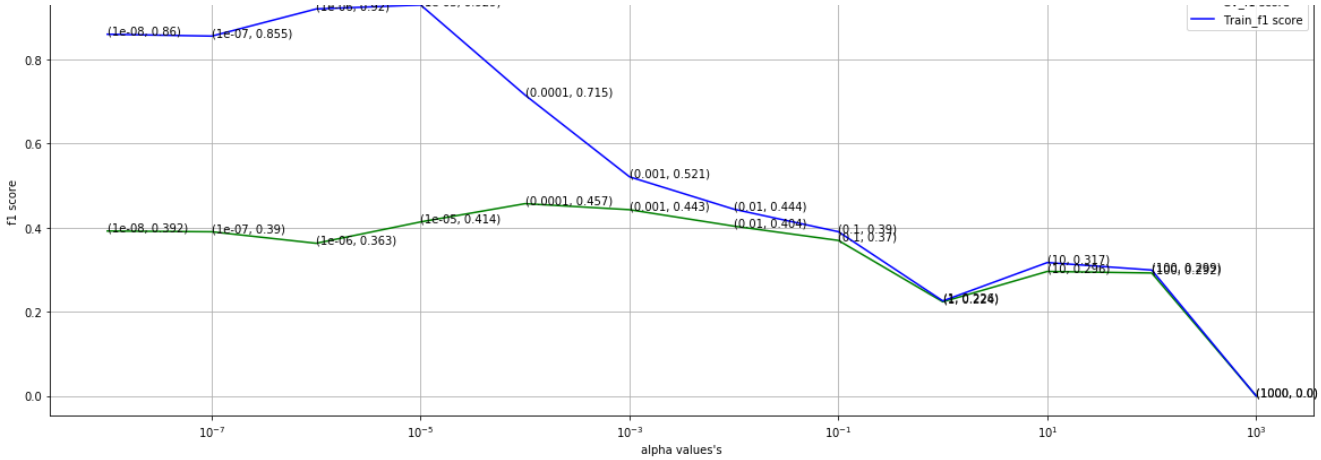
4.5.8 Char5 grams with 5 tags

In [115]:

```
lr(x_train_tfidf5,y_tain,x_cv_tfidf5,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.39214145383104126
The f1 score of cv data for alpa  1e-07 is 0.39035679845708776
The f1 score of cv data for alpa  1e-06 is 0.36294092647683807
The f1 score of cv data for alpa  1e-05 is 0.4137931034482759
The f1 score of cv data for alpa  0.0001 is 0.45716451857467777
The f1 score of cv data for alpa  0.001 is 0.4427949007584315
The f1 score of cv data for alpa  0.01 is 0.40371643120784007
The f1 score of cv data for alpa  0.1 is 0.36971667440780304
The f1 score of cv data for alpa  1 is 0.22381049622217683
The f1 score of cv data for alpa  10 is 0.29605866177818513
The f1 score of cv data for alpa  100 is 0.29231785163988555
The f1 score of cv data for alpa  1000 is 0.0
```





By looking above plot we can say alpha=0.001 we are getting better results

In [117]:

```
best_lr(x_train_tfidf5,y_tain,x_test_tfidf5,y_test,0.001)
```

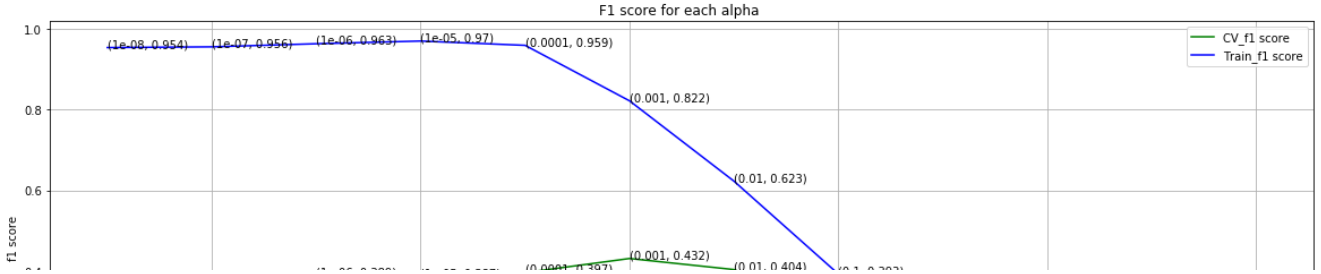
```
For values of c =  0.001 The train f1  score is: 0.5200217290082261
For values of  c =  0.001 The test  f1 score is: 0.4248756218905473
Hamming loass on test data is  0.31338529312097596
Precision recall report :
               precision    recall  f1-score   support

            0       0.20      0.58      0.30       320
            1       0.26      0.65      0.37       522
            2       0.49      0.72      0.58       871
            3       0.36      0.68      0.47       573
            4       0.20      0.55      0.29       305

   micro avg       0.31      0.66      0.42      2591
   macro avg       0.30      0.64      0.40      2591
weighted avg       0.34      0.66      0.44      2591
 samples avg       0.20      0.33      0.23      2591
```

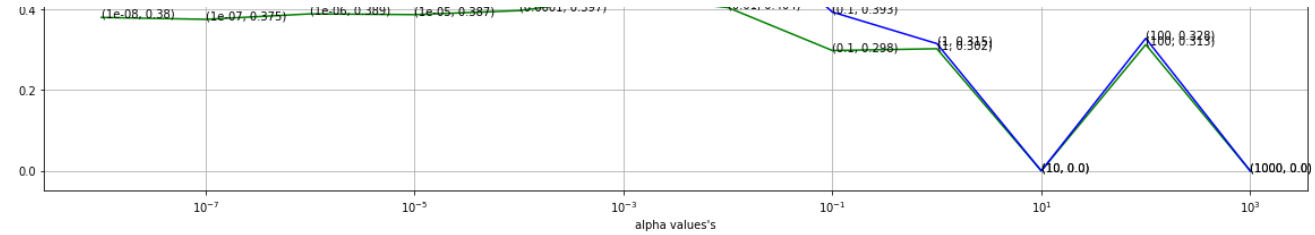
4.5.8 Uni+Bi+Tri with 5 tags

In [118]:

```
lr(x_train_tfidf123,y_tain,x_cv_tfidf123,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.37986642775696366
The f1 score of cv data for alpa  1e-07 is 0.3750809061488673
The f1 score of cv data for alpa  1e-06 is 0.3886069733180553
The f1 score of cv data for alpa  1e-05 is 0.386781475447987
The f1 score of cv data for alpa  0.0001 is 0.3966597077244259
The f1 score of cv data for alpa  0.001 is 0.43162751677852346
The f1 score of cv data for alpa  0.01 is 0.4042279674332238
The f1 score of cv data for alpa  0.1 is 0.2978008124387169
The f1 score of cv data for alpa  1 is 0.3023385923688039
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.31256505576208177
The f1 score of cv data for alpa  1000 is 0.0
```





By looking above plot we can say alpha=0.01 we are getting better results

In [120]:

```
best_lr(x_train_tfidf123,y_tain,x_test_tfidf123,y_test,0.01)
```

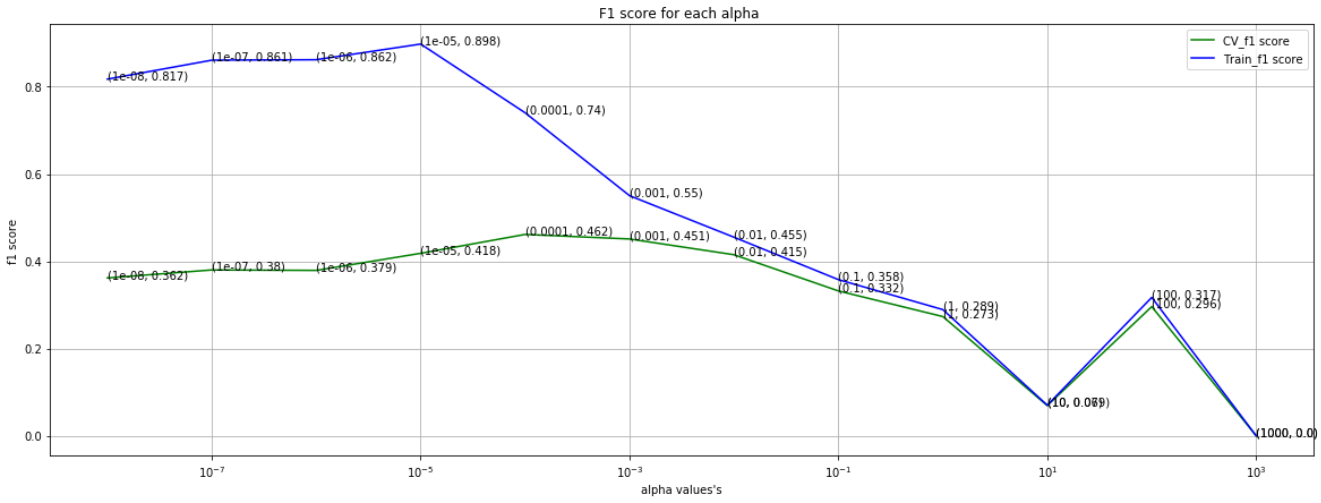
```
For values of c =  0.01 The train f1  score is: 0.6228321400159109
For values of  c =  0.01 The test  f1 score is: 0.3997655334114889
Hamming loass on test data is  0.34700101660454086
Precision recall report :
               precision    recall  f1-score   support

           0       0.18      0.71      0.28       320
           1       0.22      0.80      0.35       522
           2       0.49      0.68      0.57       871
           3       0.36      0.59      0.45       573
           4       0.21      0.42      0.28       305

   micro avg       0.29      0.66      0.40      2591
   macro avg       0.29      0.64      0.39      2591
weighted avg       0.34      0.66      0.43      2591
 samples avg       0.22      0.33      0.24      2591
```

4.5.9 Char3+Char5 grams with 5 tags

In [121]:

```
lr(xtrain_train_tfidf45,y_tain,xtrain_cv_tfidf45,y_cv)
```

```
The f1 score of cv data for alpa  1e-08 is 0.36175924014040883
The f1 score of cv data for alpa  1e-07 is 0.38012316840093435
The f1 score of cv data for alpa  1e-06 is 0.3790987941612016
The f1 score of cv data for alpa  1e-05 is 0.41825095057034223
The f1 score of cv data for alpa  0.0001 is 0.4615384615384616
The f1 score of cv data for alpa  0.001 is 0.45109395109395106
The f1 score of cv data for alpa  0.01 is 0.41500000000000004
The f1 score of cv data for alpa  0.1 is 0.33201504774766083
The f1 score of cv data for alpa  1 is 0.27330063069376315
The f1 score of cv data for alpa  10 is 0.06928406466512703
The f1 score of cv data for alpa  100 is 0.29605866177818513
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.0001 we are getting better results

In [123]:

```
best_lr(xtrain_train_tfidf45,y_tain,xtrain_test_tfidf45,y_test,0.0001)
```

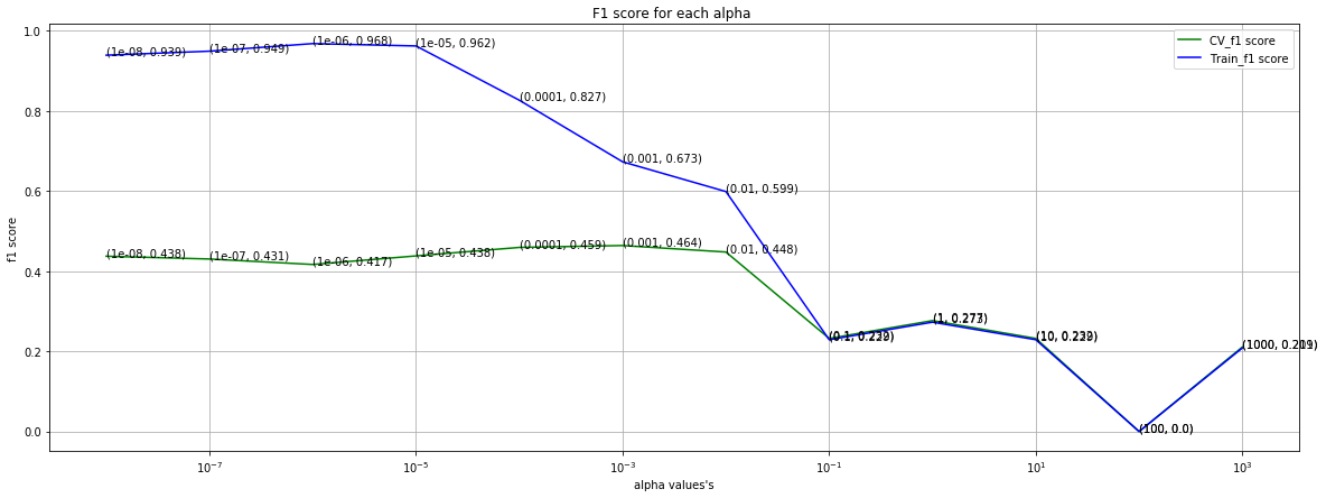
```
For values of c =  0.0001 The train f1  score is: 0.7469101641763511
For values of  c =  0.0001 The test  f1 score is: 0.42729172887429084
Hamming loass on test data is  0.2599796679091833
Precision recall report :
              precision    recall  f1-score   support

           0       0.24      0.42      0.31       320
           1       0.33      0.48      0.39       522
           2       0.49      0.67      0.57       871
           3       0.43      0.50      0.47       573
           4       0.19      0.57      0.28       305

   micro avg       0.35      0.55      0.43      2591
   macro avg       0.34      0.53      0.40      2591
weighted avg       0.38      0.55      0.44      2591
 samples avg       0.21      0.28      0.22      2591
```

4.5.10 Ngrams with 3 tags

In [125]:

```
lr(x_train_tfidf_th,y_tain1,x_cv_tfidf_th,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.437515589922674
The f1 score of cv data for alpa  1e-07 is 0.4305484366991287
The f1 score of cv data for alpa  1e-06 is 0.4168945609237314
The f1 score of cv data for alpa  1e-05 is 0.4382965495803544
The f1 score of cv data for alpa  0.0001 is 0.4594739667203435
The f1 score of cv data for alpa  0.001 is 0.4641737032569361
The f1 score of cv data for alpa  0.01 is 0.44796650717703346
The f1 score of cv data for alpa  0.1 is 0.23229750382068262
The f1 score of cv data for alpa  1 is 0.2767615714967393
The f1 score of cv data for alpa  10 is 0.23229750382068262
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.2109016811003566
```



By looking above plot we can say alpha=0.01 we are getting better results

In [126]:

```
best_lr(x_train_tfidf_th,y_tain1,x_test_tfidf_th,y_test1,0.01)
```

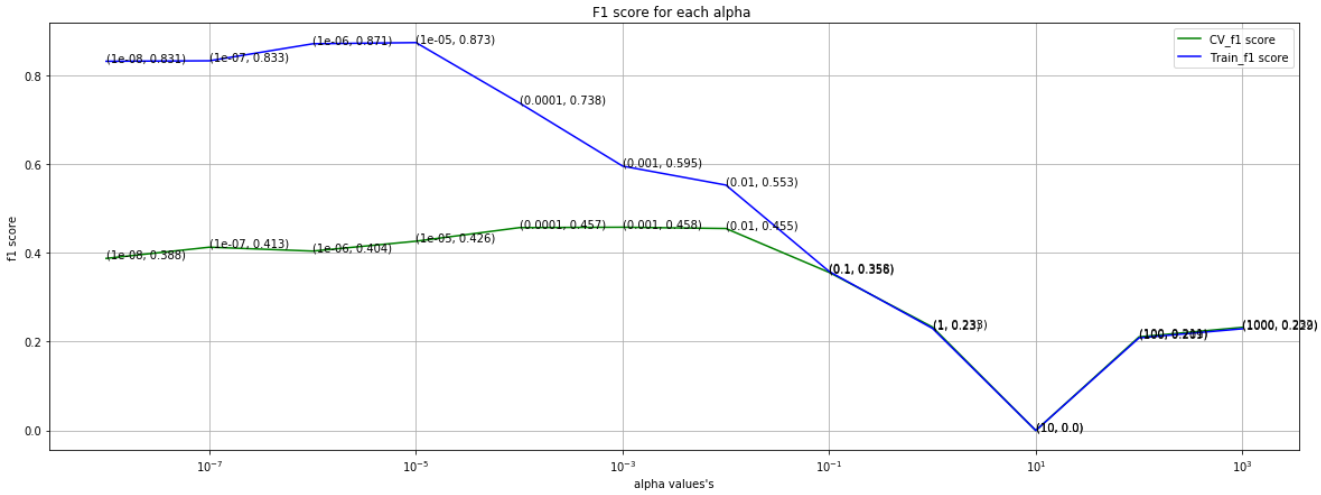
```
For values of c =  0.01 The train f1  score is: 0.543373090003353
For values of  c =  0.01 The test  f1 score is: 0.4559721011333914
Hamming loass on test data is  0.42290748898678415
Precision recall report :
              precision    recall  f1-score   support

           0       0.26      0.52      0.34       526
           1       0.37      0.93      0.53       868
           2       0.29      0.87      0.43       567

   micro avg       0.32      0.80      0.46      1961
   macro avg       0.31      0.77      0.44      1961
weighted avg       0.32      0.80      0.45      1961
 samples avg       0.25      0.35      0.27      1961
```

4.5.11 unigrams with 3 tags

In [127]:

```
lr(x_train_tfidf_th1,y_tain1,x_cv_tfidf_th1,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.3875375785733807
The f1 score of cv data for alpa  1e-07 is 0.4128205128205128
The f1 score of cv data for alpa  1e-06 is 0.40400216333153055
The f1 score of cv data for alpa  1e-05 is 0.42622950819672134
The f1 score of cv data for alpa  0.0001 is 0.45681281176161725
The f1 score of cv data for alpa  0.001 is 0.45774303859306303
The f1 score of cv data for alpa  0.01 is 0.45487364620938625
The f1 score of cv data for alpa  0.1 is 0.35601828339258507
The f1 score of cv data for alpa  1 is 0.23289070480081717
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.2109016811003566
The f1 score of cv data for alpa  1000 is 0.23229750382068262
```



By looking above plot we can say alpha=0.01 we are getting better results

In [130]:

```
best_lr(x_train_tfidf_th1,y_tain1,x_test_tfidf_th1,y_test1,0.01)
```

```
For values of c =  0.01 The train f1  score is: 0.4842461785284549
For values of  c =  0.01 The test  f1 score is: 0.44930584770719395
Hamming loass on test data is  0.4435784479837343
Precision recall report :
              precision    recall  f1-score   support

           0       0.20      0.96      0.33       526
           1       0.45      0.82      0.58       868
           2       0.36      0.68      0.47       567
```

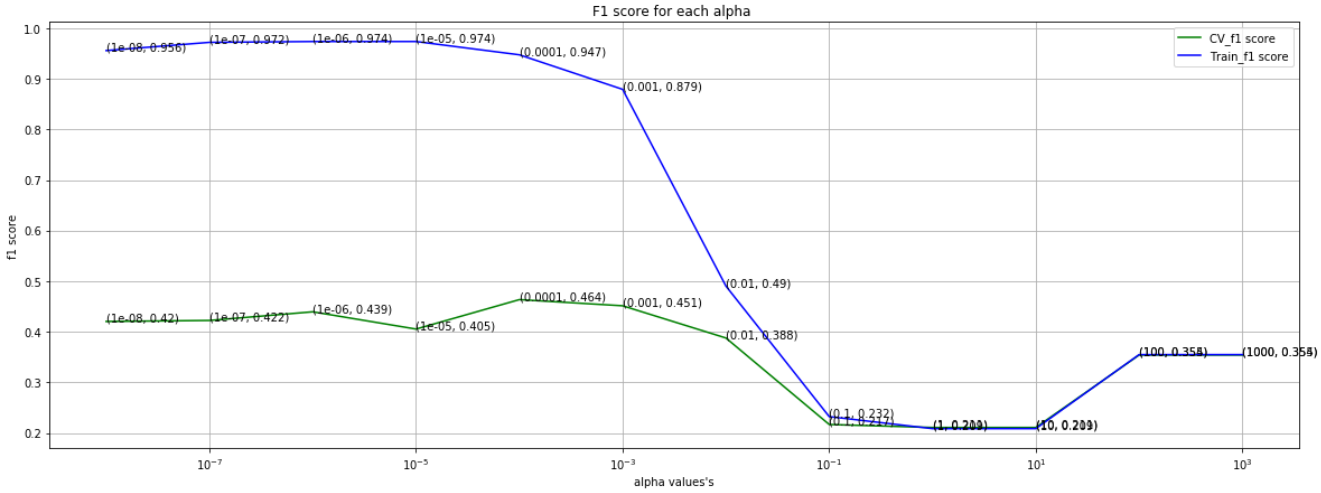
```
   micro avg       0.31      0.82      0.45      1961
   macro avg       0.34      0.82      0.46      1961
weighted avg       0.36      0.82      0.48      1961
 samples avg       0.25      0.36      0.28      1961
```

4.5.12 Bigrams with 3 tags

In [131]:

```
lr(x_train_tfidf_th2,y_tain1,x_cv_tfidf_th2,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.42045005921831813
The f1 score of cv data for alpa  1e-07 is 0.4224211423699915
The f1 score of cv data for alpa  1e-06 is 0.4394985929905346
The f1 score of cv data for alpa  1e-05 is 0.4054054054054054
The f1 score of cv data for alpa  0.0001 is 0.4636015325670499
The f1 score of cv data for alpa  0.001 is 0.4514301897479468
The f1 score of cv data for alpa  0.01 is 0.38775510204081637
The f1 score of cv data for alpa  0.1 is 0.21681864235055726
The f1 score of cv data for alpa  1 is 0.2109016811003566
The f1 score of cv data for alpa  10 is 0.2109016811003566
The f1 score of cv data for alpa  100 is 0.35404992358634746
The f1 score of cv data for alpa  1000 is 0.35404992358634746
```



By looking above plot we can say alpha=0.01 we are getting better results

In [136]:

```
best_lr(x_train_tfidf_th2,y_tain1,x_test_tfidf_th2,y_test1,0.01)
```

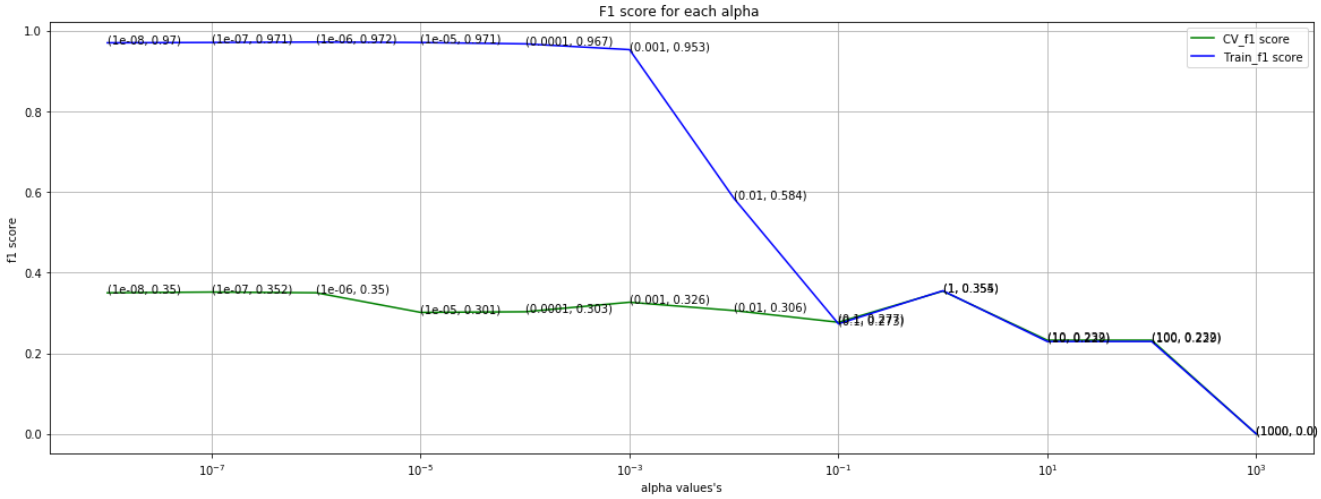
```
For values of c =  0.01 The train f1  score is: 0.41644241119483316
For values of  c =  0.01 The test  f1 score is: 0.37518405811328165
Hamming loass on test data is  0.7189653224895516
Precision recall report :
              precision    recall  f1-score   support

           0       0.19      0.97      0.32       526
           1       0.32      0.96      0.48       868
           2       0.19      1.00      0.32       567

   micro avg       0.23      0.97      0.38      1961
   macro avg       0.23      0.98      0.37      1961
weighted avg       0.25      0.97      0.39      1961
 samples avg       0.22      0.43      0.28      1961
```

4.5.13 Tri Grams with 3 tags

In [137]:

```
lr(x_train_tfidf_th3,y_tain1,x_cv_tfidf_th3,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.3497933434848815
The f1 score of cv data for alpa  1e-07 is 0.3516629711751663
The f1 score of cv data for alpa  1e-06 is 0.3499214248297538
The f1 score of cv data for alpa  1e-05 is 0.3012232415902141
The f1 score of cv data for alpa  0.0001 is 0.302803738317757
The f1 score of cv data for alpa  0.001 is 0.32640332640332637
The f1 score of cv data for alpa  0.01 is 0.30576441102756896
The f1 score of cv data for alpa  0.1 is 0.2767615714967393
The f1 score of cv data for alpa  1 is 0.35404992358634746
The f1 score of cv data for alpa  10 is 0.23229750382068262
The f1 score of cv data for alpa  100 is 0.23229750382068262
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.01 we are getting better results

In [139]:

```
best_lr(x_train_tfidf_th3,y_tain1,x_test_tfidf_th3,y_test1,0.01)
```

```
For values of c =  0.01 The train f1  score is: 0.518320132201265
For values of  c =  0.01 The test  f1 score is: 0.3705417389627916
Hamming loass on test data is  0.36880153620241723
Precision recall report :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       526
           1       0.29      1.00      0.45       868
           2       0.34      0.16      0.22       567

   micro avg       0.30      0.49      0.37      1961
   macro avg       0.21      0.39      0.23      1961
weighted avg       0.23      0.49      0.27      1961
 samples avg       0.29      0.22      0.24      1961
```

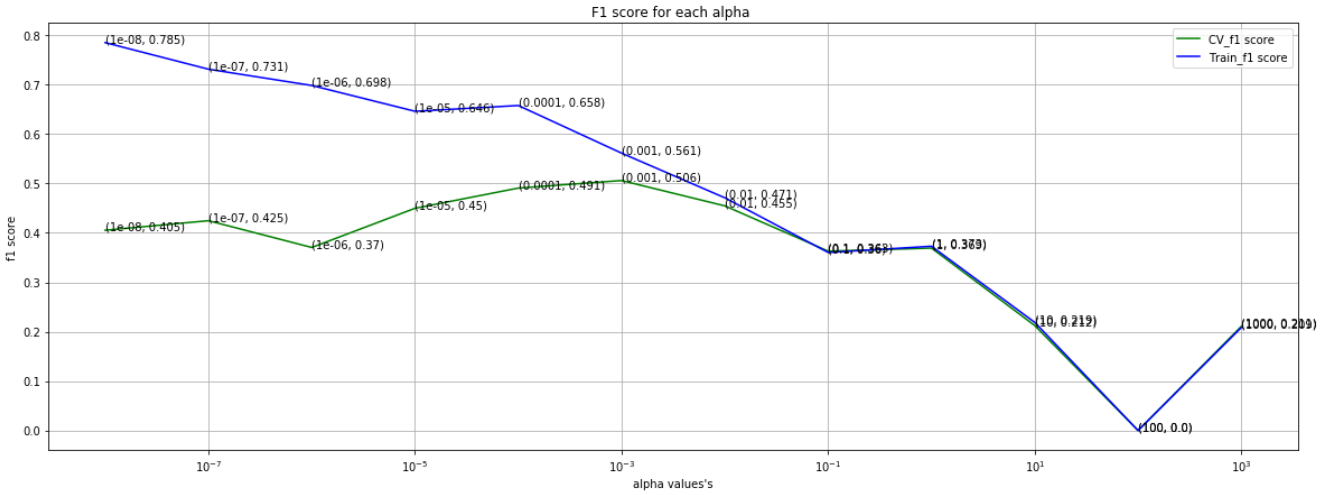
4.5.14 char 3 with 3 tags

In [140]:

```
lr(x_train_tfidf_th4,y_tain1,x_cv_tfidf_th4,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.40534262485482
The f1 score of cv data for alpa  1e-07 is 0.42467378410438905
The f1 score of cv data for alpa  1e-06 is 0.37048917401764236
The f1 score of cv data for alpa  1e-05 is 0.4498297616663328
The f1 score of cv data for alpa  0.0001 is 0.49069709749441826
```

```
The f1 score of cv data for alpa  0.001 is 0.5059802392095684
The f1 score of cv data for alpa  0.01 is 0.45464061409630147
The f1 score of cv data for alpa  0.1 is 0.3629827206308709
The f1 score of cv data for alpa  1 is 0.369222548757681
The f1 score of cv data for alpa  10 is 0.21237244897959187
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.2109016811003566
```



By looking above plot we can say alpha=0.001 we are getting better results

In [141]:

```
best_lr(x_train_tfidf_th4,y_tain1,x_test_tfidf_th4,y_test1,0.001)
```

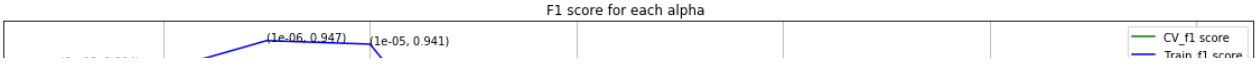
```
For values of c =  0.001 The train f1  score is: 0.5332179930795848
For values of  c =  0.001 The test  f1 score is: 0.4912216532553036
Hamming loass on test data is  0.3142437591776799
Precision recall report :
              precision    recall  f1-score   support

           0       0.29      0.56      0.38       526
           1       0.53      0.70      0.60       868
           2       0.33      0.77      0.46       567

   micro avg       0.38      0.68      0.49      1961
   macro avg       0.38      0.68      0.48      1961
weighted avg       0.41      0.68      0.50      1961
 samples avg       0.21      0.29      0.23      1961
```

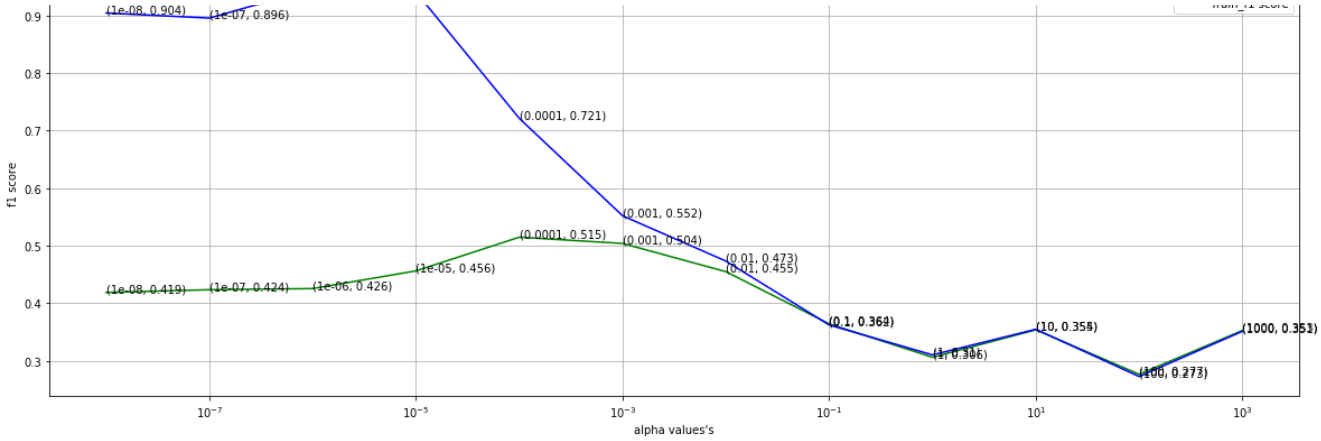
4.5.15 Char 5 with 3 tags

In [142]:

```
lr(x_train_tfidf_th5,y_tain1,x_cv_tfidf_th5,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.41894674904383644
The f1 score of cv data for alpa  1e-07 is 0.4236877523553163
The f1 score of cv data for alpa  1e-06 is 0.4256542996836353
The f1 score of cv data for alpa  1e-05 is 0.45643153526970953
The f1 score of cv data for alpa  0.0001 is 0.5149253731343284
The f1 score of cv data for alpa  0.001 is 0.5040829838887663
The f1 score of cv data for alpa  0.01 is 0.4550337896378444
The f1 score of cv data for alpa  0.1 is 0.3643564356435644
The f1 score of cv data for alpa  1 is 0.30626826891847997
The f1 score of cv data for alpa  10 is 0.35404992358634746
The f1 score of cv data for alpa  100 is 0.27676157149673 93
The f1 score of cv data for alpa  1000 is 0.3527914744711309
```





By looking above plot we can say alpha=0.001 we are getting better results

In [144]:

```
best_lr(x_train_tfidf_th5,y_tain1,x_test_tfidf_th5,y_test1,0.001)
```

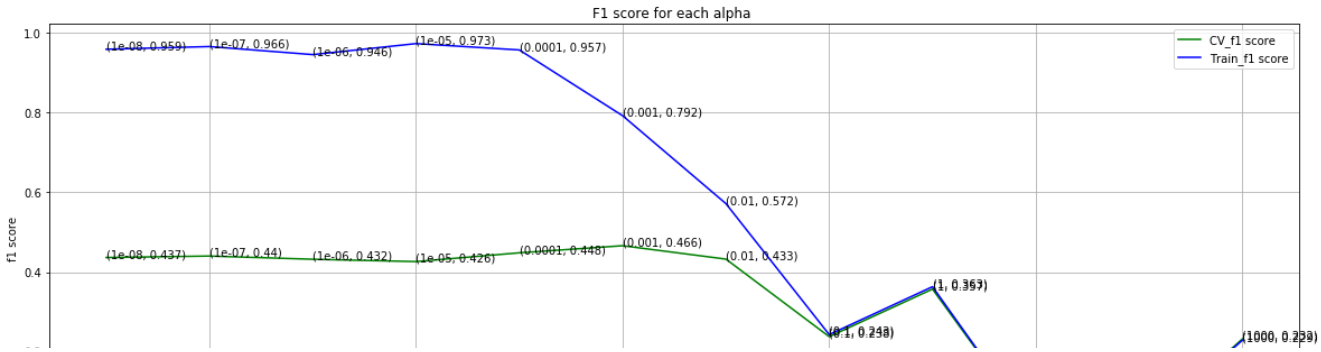
```
For values of c =  0.001 The train f1  score is: 0.5645447029644501
For values of  c =  0.001 The test  f1 score is: 0.5038541079150216
Hamming loass on test data is  0.29809104258443464
Precision recall report :
              precision    recall  f1-score   support

           0       0.28      0.59      0.38       526
           1       0.53      0.72      0.61       868
           2       0.38      0.70      0.49       567

   micro avg       0.40      0.68      0.50      1961
   macro avg       0.40      0.67      0.49      1961
weighted avg       0.42      0.68      0.52      1961
 samples avg       0.21      0.29      0.23      1961
```

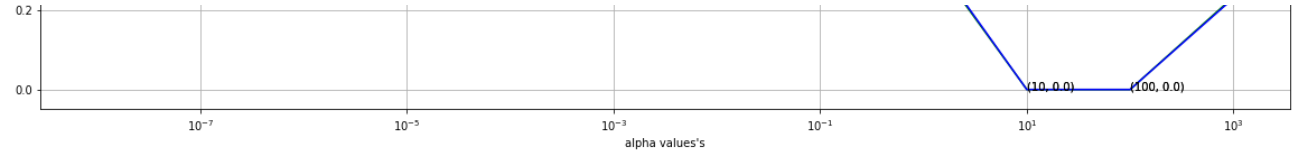
4.5.16 Bi+Tri+Uni Grams with 3 tags

In [145]:

```
lr(x_train_tfidf_th123,y_tain1,x_cv_tfidf_th123,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.43655764323822777
The f1 score of cv data for alpa  1e-07 is 0.44022770398481975
The f1 score of cv data for alpa  1e-06 is 0.4322217410134257
The f1 score of cv data for alpa  1e-05 is 0.42634425707922363
The f1 score of cv data for alpa  0.0001 is 0.44836433611289 29
The f1 score of cv data for alpa  0.001 is 0.46614931763446615
The f1 score of cv data for alpa  0.01 is 0.43258219431104544
The f1 score of cv data for alpa  0.1 is 0.23799690242643262
The f1 score of cv data for alpa  1 is 0.35728744939271256
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.0
The f1 score of cv data for alpa  1000 is 0.23229750382068262
```





By looking above plot we can say alpha=0.01 we are getting better results

In [146]:

```
best_lr(x_train_tfidf_th123,y_tain1,x_test_tfidf_th123,y_test1,0.01)
```

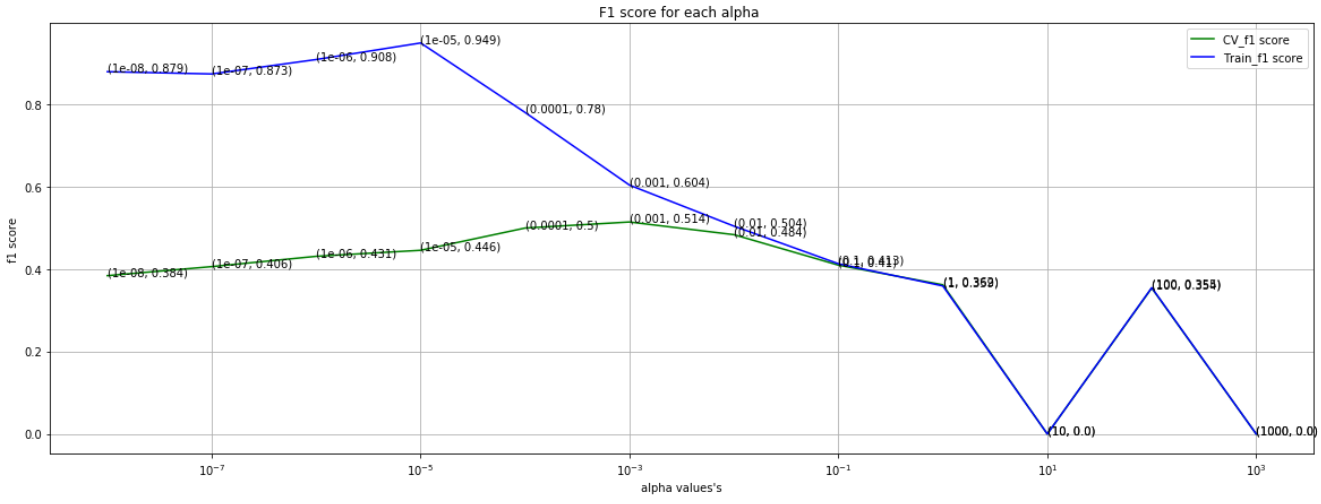
```
For values of c =  0.01 The train f1  score is: 0.650925051391744
For values of  c =  0.01 The test  f1 score is: 0.4755244755244755
Hamming loass on test data is  0.3558115892917655
Precision recall report :
              precision    recall  f1-score   support

           0       0.24      0.67      0.35       526
           1       0.43      0.83      0.57       868
           2       0.38      0.63      0.48       567

   micro avg       0.35      0.73      0.48      1961
   macro avg       0.35      0.71      0.47      1961
weighted avg       0.37      0.73      0.49      1961
 samples avg       0.24      0.32      0.26      1961
```

4.5.17 char3+char5 grams with 3 tags

In [147]:

```
lr(x_train_tfidf_th45,y_tain1,x_cv_tfidf_th45,y_cv1)
```

```
The f1 score of cv data for alpa  1e-08 is 0.3840665873959572
The f1 score of cv data for alpa  1e-07 is 0.4060324825986079
The f1 score of cv data for alpa  1e-06 is 0.43101111412306864
The f1 score of cv data for alpa  1e-05 is 0.44565217391304346
The f1 score of cv data for alpa  0.0001 is 0.4996137007468452
The f1 score of cv data for alpa  0.001 is 0.5143004380314352
The f1 score of cv data for alpa  0.01 is 0.4835296466360551
The f1 score of cv data for alpa  0.1 is 0.4096113074204947
The f1 score of cv data for alpa  1 is 0.36193339500462535
The f1 score of cv data for alpa  10 is 0.0
The f1 score of cv data for alpa  100 is 0.35404992358634746
The f1 score of cv data for alpa  1000 is 0.0
```



By looking above plot we can say alpha=0.001 we are getting better results

In [149]:

```
best_lr(x_train_tfidf_th45,y_tain1,x_test_tfidf_th45,y_test1,0.001)
```

```
For values of c =  0.001 The train f1  score is: 0.5953452977360644
For values of  c =  0.001 The test  f1 score is: 0.516921837228042
Hamming loass on test data is  0.27086863210211226
Precision recall report :
               precision    recall  f1-score   support

           0       0.36      0.39      0.38       526
           1       0.52      0.76      0.61       868
           2       0.36      0.74      0.49       567

   micro avg       0.43      0.65      0.52      1961
   macro avg       0.41      0.63      0.49      1961
weighted avg       0.43      0.65      0.51      1961
 samples avg       0.22      0.27      0.23      1961
```

Performance table

In [35]:

```
from prettytable import PrettyTable

x = PrettyTable()

x.field_names = ["Model", "Vectorizer", " 5tags_micro f1score", " 3tags_micro f1score"]
x.add_row(["OVR With Logistic Regression", 'Bag Of Words',0.406,0.467])
x.add_row(["OVR With Logistic Regression",'N Grams', 0.388,0.45])
x.add_row(["OVR With Logistic Regression",'Uni Grams', 0.406,0.44])
x.add_row(["OVR With Logistic Regression",'Bi Grams', 0.34,0.37])
x.add_row(["OVR With Logistic Regression",'Tri Grams', 0.32,0.37])
x.add_row(["OVR With Logistic Regression",'Char3 Grams', 0.42,0.49])
x.add_row(["OVR With Logistic Regression",'Char5 Grams', 0.424,0.503])
x.add_row(["OVR With Logistic Regression",'Uni+Bi+Tri Grams', 0.39,0.47])
x.add_row(["OVR With Logistic Regression",'Char3+Char5 grams', 0.427,0.5169])
x.add_row(["OVR With SVM", 'Bag Of Words',0.404,0.467])
x.add_row(["OVR With SVM",'N Grams', 0.398,0.456])
x.add_row(["OVR With SVM",'Uni Grams', 0.395,0.46])
x.add_row(["OVR With SVM",'Bi Grams', 0.383,0.23])
x.add_row(["OVR With SVM",'Tri Grams', 0.396,0.35])
x.add_row(["OVR With SVM",'Char3 Grams', 0.417,0.503])
x.add_row(["OVR With SVM",'Char5 Grams', 0.4253,0.507])
x.add_row(["OVR With SVM",'Uni+Bi+Tri Grams', 0.349,0.469])
x.add_row(["OVR With SVM",'Char3+Char5 grams', 0.4,0.515])
print(x)
```

Model	Vectorizer	5tags_micro f1score	3tags_micro f1score
OVR With Logistic Regression	Bag Of Words	0.406	0.467
OVR With Logistic Regression	N Grams	0.388	0.45
OVR With Logistic Regression	Uni Grams	0.406	0.44
OVR With Logistic Regression	Bi Grams	0.34	0.37
OVR With Logistic Regression	Tri Grams	0.32	0.37
OVR With Logistic Regression	Char3 Grams	0.42	0.49
OVR With Logistic Regression	Char5 Grams	0.424	0.503
OVR With Logistic Regression	Uni+Bi+Tri Grams	0.39	0.47
OVR With Logistic Regression	Char3+Char5 grams	0.427	0.5169
OVR With SVM	Bag Of Words	0.404	0.467
OVR With SVM	N Grams	0.398	0.456
OVR With SVM	Uni Grams	0.395	0.46
OVR With SVM	Bi Grams	0.383	0.23
OVR With SVM	Tri Grams	0.396	0.35
OVR With SVM	Char3 Grams	0.417	0.503
OVR With SVM	Char5 Grams	0.4253	0.507
OVR With SVM	Uni+Bi+Tri Grams	0.349	0.469
OVR With SVM	Char3+Char5 grams	0.4	0.515

Observations:

1) from pretty table we can see that f1_score is better when we use 3 tags .
2) Logistic regression with Char3+achr5 and 3 tags we are getting best f1score of 0.5169
3) In the whole training process best hyperparametr is taken in such a way that the model is neither overfitted nor under fitted
4) Idea of feature engineering is inspired from the reasearch paper (Mentioned in 1.2)
5) For multi class multi label classification we can also use other techiniques like Problem Transformation Method,Adapted Algorithm Method,Ensemble Approaches but due to computing challenges we have used One Vs rest(link to the article Mentioned in 1.2)
6) In One VS rest approach we can also fine tune the probablity threshold values as well . I have tuned them but got bad results so left as is(0.5).
7) By Doing EDA we have got some insights like most popular tag,avg tag per question,how many times tags appeared etc