In [5]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import re
import nltk
from nltk import FreqDist
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords, wordnet
from nltk.stem import WordNetLemmatizer
from wordcloud import WordCloud
from collections import Counter
from itertools import chain
import contractions
from nltk.collocations import BigramAssocMeasures, BigramCollocationFinde
from nltk.collocations import TrigramAssocMeasures, TrigramCollocationFin
from nltk import ngrams
```

In [6]:
```
data = pd.read_csv("dataset/mb_data.csv")
data.head()
```

Out[6]:

	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8675 entries, 0 to 8674
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   type    8675 non-null   object
 1   posts   8675 non-null   object
dtypes: object(2)
memory usage: 135.7+ KB
```

In [8]:
```
data.describe()
```

Out[8]:

	type	posts
count	8675	8675
unique	16	8675
top	INFP	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
freq	1832	1

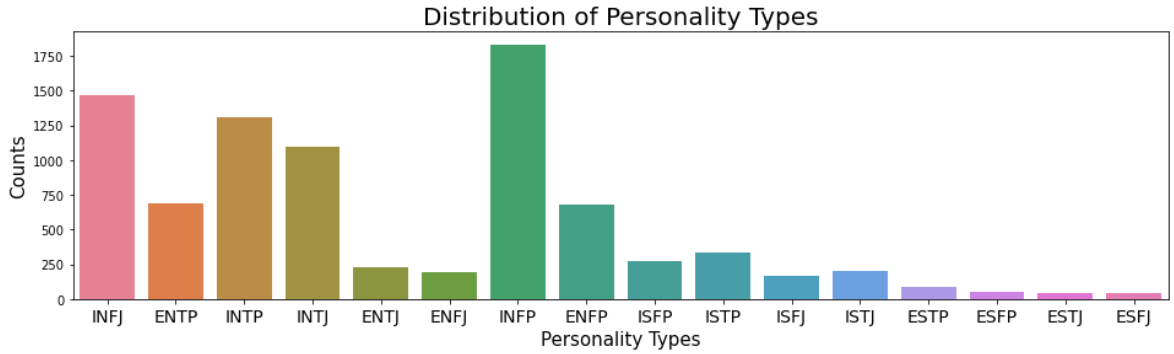
In [9]:
```
_classes = data.type.unique()
```

```
print(_classes)
```

```
['INFJ' 'ENTP' 'INTP' 'INTJ' 'ENTJ' 'ENFJ' 'INFP' 'ENFP' 'ISFP' 'ISTP'
 'ISFJ' 'ISTJ' 'ESTP' 'ESFP' 'ESTJ' 'ESFJ']
```

In [10]:
```
def show_class_distribution(data, x="type", figsize=(16,4), title="Distri
    plt.figure(figsize=figsize)
    sns.countplot(x=x, data=data, palette=palette)
    plt.xlabel("Personality Types", size=15)
    plt.ylabel("Counts", size=15)
    plt.xticks(size=xticks_size)
    plt.title(title, size=20)
    plt.show()
```

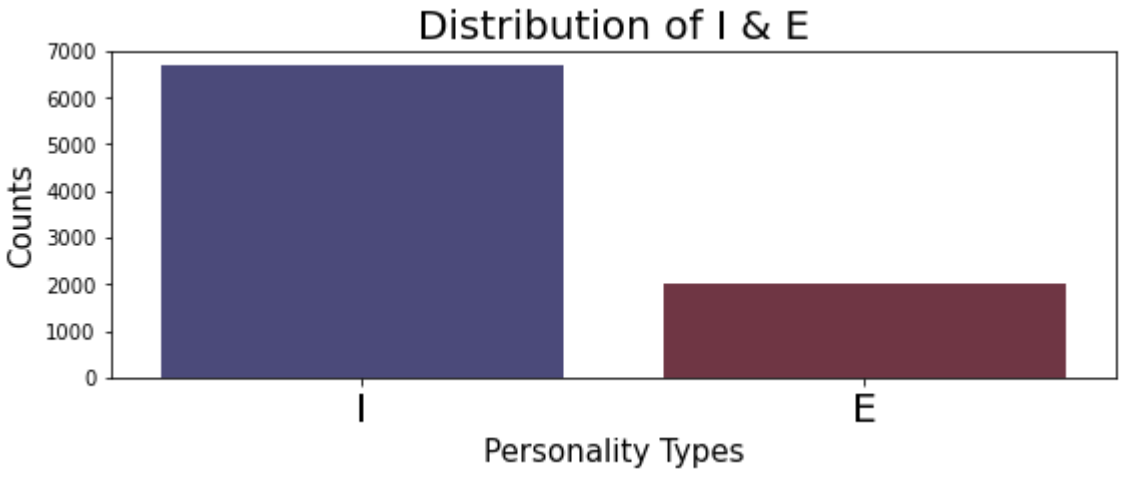
In [11]:
```
show_class_distribution(data, xticks_size=14)
```



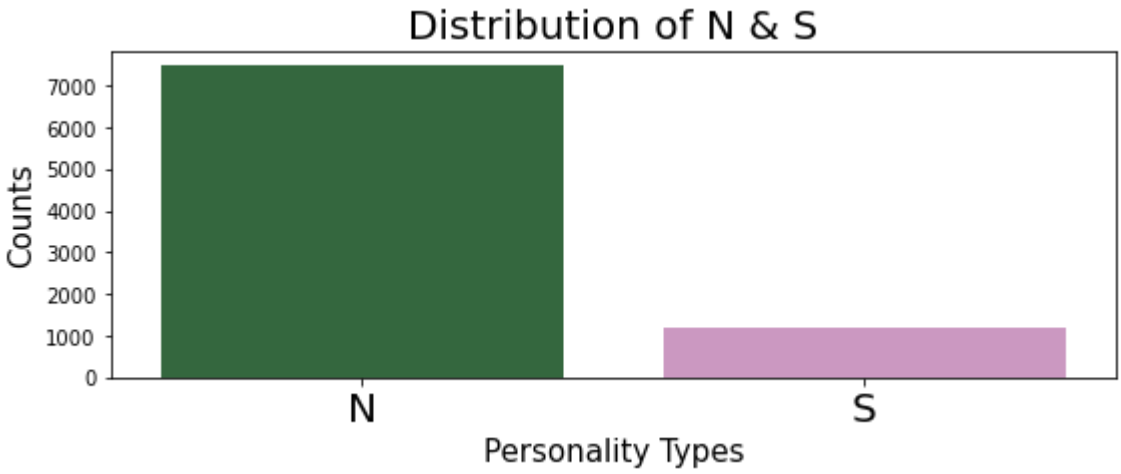
In [12]:
```
def divide_types(df):
    df["E-I"] = ""
    df["N-S"] = ""
    df["F-T"] = ""
    df["J-P"] = ""
    for index, row in df.iterrows():
        row["E-I"] = "E" if row.type[0] == "E" else "I"
        row["N-S"] = "N" if row.type[1] == "N" else "S"
        row["F-T"] = "F" if row.type[2] == "F" else "T"
        row["J-P"] = "J" if row.type[3] == "J" else "P"
    return df

data = divide_types(data)
```

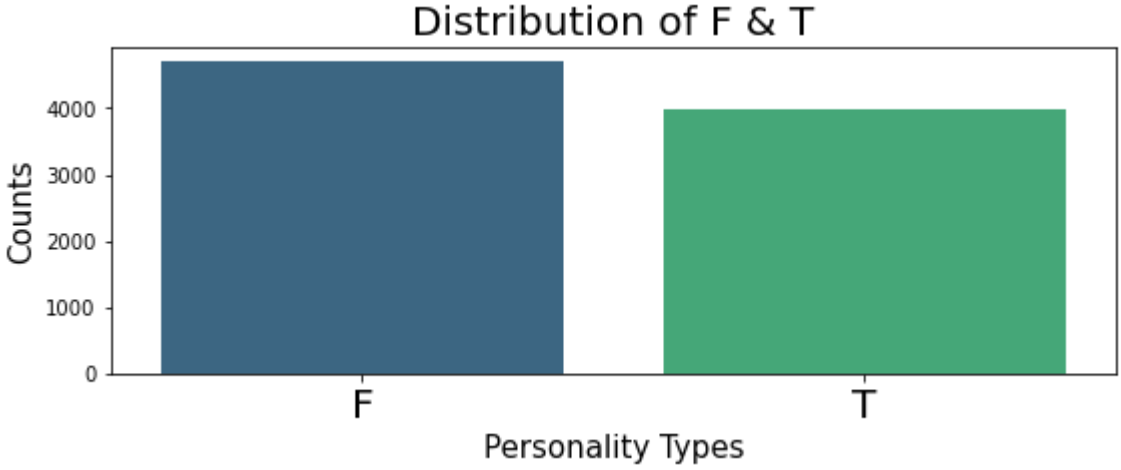
In [13]:
```
show_class_distribution(data, x="E-I", title="Distribution of I & E", fig
```



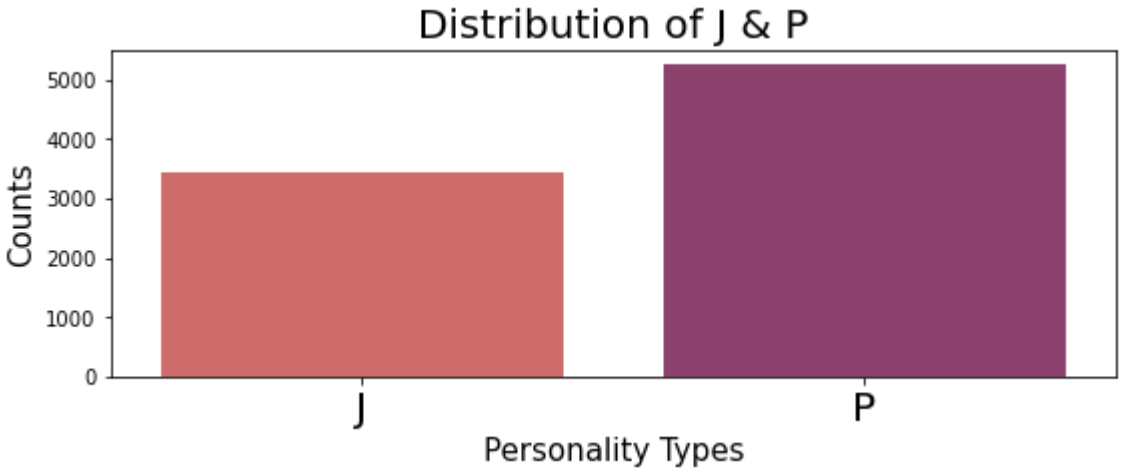
In [14]:
```
show_class_distribution(data, x="N-S", title="Distribution of N & S", fig
```



In [15]:
```
show_class_distribution(data, x="F-T", title="Distribution of F & T", fig
```



In [16]:
```
show_class_distribution(data, x="J-P", title="Distribution of J & P", fig
```



sample post

In [17]:
```
data.loc[7, "posts"]
```

Out[17]: "'I tend to build up a collection of things on my desktop that i use fre
quently and then move them into a folder called 'Everything' from there
it get sorted into type and sub type|||i ike to collect odd objects, eve
n at work...a lot of people would call it junk but i like to collect it.
Old unused software? ill take that off your hands :) i have a bunch of o
ld adobe...|||i think its quite normal, i tend to only see my friends in
real life every couple of months, as said earlier some people just dont
get it but the good ones do :) Edit: i mostly mean tolerate it...|||whe
re do we go when we sleep? is dreaming another form of being awake? how
many more layers of this are there if any? thoughts about sleep keep m
e up at night Edit: sometimes im too scared...|||thanks|||i wish i was
free to follow my interests as i desired i feel as though wishes are me
ant for impossible things|||by seeing do you mean visual interpreting or
seeing as in mentally understanding the concept?|||hello|||i feel as tho
ugh i am incapable of creating anything and i wish i could|||i cant st
and the interviewer christ that laugh... is he intj? hmmm it would be
interesting to see an intj on this show, i doubt they would be that inte
resting to the general public though ...|||know yourself and be yourself
|||Do you think Fi or Fe sounds more like me? which one do you think so
unds like you? Why do you require input from others to know what you ar
e?|||Question: do INTJs lean more towards Alternative Rock then other ty
pes of music? And if so, why? My Answer: well, if you went through all
the pages and then sorted all the songs by genre/style...|||sometimes i
look at people and i see them , well on the outside at least, doing all
these things and saying all these words and i wonder what it would be li
ke to act that way... am i missing out on...|||a lounge huh? what does o
ne do in a lounge? or is it best not to define it and just enjoy it as
is?|||Do it|||went on holiday for just over a month, thought things woul
d change. How naive of me.. feels nice to browse back on this forum her
e though, its been a while since i surrounded myself with somewhat...|||
yes i would say i am, more than physical appearance to an certain degre
e. what are they? i am unsure they just generally cant be a terrible pe
rson by my standards|||i like to lurk, in my case at least its mostly be
cause i tend to believe i have nothing interesting to contribute to the
conversation so why add anything? logging out to purposely lurk seems l
ike...|||i think id wait for him to swing first before taking further ac
tion but i would not encourage them to take a swing a part of me wants
to fight him though...|||would you say there is complexity in simplicit
y?|||I normally vote for whoever amuses me the most.. perhaps one day
i'll care more Edit: other than what amuses me at the time, ill vote fo
r whatever would apparently benefit me the most....|||long distance is h
ard|||INTJ's and what effects their sanity levels Mental illness|||So i
think about which thoughts i wish to express in written format, then i p
roceed to perform the physical movements necessary to express the requir
ed thoughts in whichever medium is required or...|||651762 i got this, s
eems right to me. I always score intj, extra heavy on the introversion :
3|||I like this https://www.youtube.com/watch?v=e4dT8FJ2GE0 I like the
sound, the content/lyrics are not too important, the sound is what i val
ue most. Is it my favourite? probably not, i heard...|||to others maybe,
although internally i am acutely aware of any mistakes i have made|||ass
uming they would listen to me, i would each give them a solo task that l
eads to the successful completion of the project how would i make them
listen to me? explain that the success of the...|||i am a fan of the ide
a that a celestial being died and its essence became everything we know,
what kind of sacrifice would have to be made to create such wondrous thi
ngs? (stars, planets, cosmic...|||Nah|||1) Since when do you need a mana
ger present to make returns?? or was it because I work there? - check th
e company policy on this 2) Why did they want to save my receipt? -not s
ure, perhaps to...|||i remember my first encounter with this type of mus
ic was BOA - Hurricane Venus, saw it played on a starcraft 2 stream roun

```
d 201011 or something and was kinda hooked. Don't follow/watch it muc
h...|||what happens when this riddle gets solved? what use is this infor
mation?  haha i am clearly not a fan of riddles  Edit thoughts: Girlfri
end asked me a few riddles and it was infuriating for...|||i kept pressi
ng them to talk about something they didnt want to, i regret it deeply||
|It is not hands that summon us. It is desire.  I want nothing to do wit
h that puzzle box. Nope. Nope. Nope|||i used to think life was dull and
everything was boring, then i realised that i havent actually done every
thing so it felt incorrect in thinking that way|||ahh yes the pain of li
ving, its beautiful isnt it?|||i got  Your score for openness was 70%.
This is in the moderate range. Your score for conscientiousness was low,
at 35%. Your score for Extraversion was low, at 15%. Your score for...||
|What do you mean by humanities? i like to think about the direction of
the human race as a species and i like to appreciate where weve come fro
m so i guess i love humanities?|||1. Attempt to think less about things
2. Be more decisive and confident in my decisions/thoughts (i cant help
but play devils advocate with myself, i guess this ties into number 1)
3. change my...|||Ahh yes that sinking feeling when something goes incre
dibly wrong or not to plan. How do i get over it? embrace the madness, h
ow are you going to deal with your scenario now with all these new...|||
I like to imagine it kinda like this 629346  except the ring is a billio
n times higher and nearly impossible to scale and then in the middle is
a beautiful garden and house where i like to chill....|||if you look at
the first sentence of most of them you can gather the feeling they are t
rying to portray, followed by a rather nice scenario.  So yes i do feel
those feels, but not for those...|||I think its the introversion crossed
with the feels that does it for me.|||first encountered mbti?  hmm was a
personality test i took in highschool, didnt care or even pay attention
to it until later. stumbled upon it somehow and was interested  Im not t
oo into it as...|||Treat Yo Self|||Question: Can INTJs be try-hards?  An
swer: sure|||No im not, our values of friendship are clearly different.
An emotional investment of any kind is a big deal to me, a HUGE deal, to
have it not appreciated and/or reciprocated is soul destroying. ...|||pe
tty? perhaps, but why bother with someone who isnt going to give me what
i want when i can just move on? what a waste of my time and emotional en
ergy.  Hopefully for you this friend of yours is...'"
```

In [18]:
```
def fix_contractions(df, column_name = "posts", new_column="cleaned_post"
    df[new_column] = df[column_name].apply(lambda x: contractions.fix(x))
    return df

data = fix_contractions(data)
```

In [19]:
```
def clean_data(df, column_name = "cleaned_post"):
    df[column_name] = df[column_name].apply(lambda x: x.lower())
    df[column_name] = df[column_name].apply(lambda x: re.sub(r'@([a-zA-Z0
    df[column_name] = df[column_name].apply(lambda x: re.sub(r'#([a-zA-Z0
    df[column_name] = df[column_name].apply(lambda x: re.sub(r'http[s]?:/
    df[column_name] = df[column_name].apply(lambda x: re.sub(r'[^A-Za-z]+
    df[column_name] = df[column_name].apply(lambda x: re.sub(r' +', ' ',
    df[column_name] = df[column_name].apply(lambda x: " ".join([word for
    return df

data = clean_data(data)
```

In [20]:
```
data.loc[7,"cleaned_post"]
```

Out[20]: 'tend build collection things desktop that use frequently and then move them into folder called everything from there get sorted into type and sub type ike collect odd objects even work lot people would call junk but like collect old unused software ill take that off your hands have bunch old adobe think its quite normal tend only see friends real life every couple months said earlier some people just not get but the good ones edit mostly mean tolerate where when sleep dreaming another form being awake how many more layers this are there any thoughts about sleep keep night edit sometimes too scared thanks wish was free follow interests desired feel though wishes are meant for impossible things seeing you mean visual interpreting seeing mentally understanding the concept hello feel though incapable creating anything and wish could cannot stand the interviewer christ that laugh intj hmmm would interesting see intj this show doubt they would that interesting the general public though know yourself and yourself you think sounds more like which one you think sounds like you why you require input from others know what you are question intjs lean more towards alternative rock then other types music and why answer well you went through all the pages and then sorted all the songs genre style sometimes look people and see them well the outside least doing all these things and saying all these words and wonder what would like act that way missing out lounge huh what does one lounge best not define and just enjoy went holiday for just over month thought things would change how naive feels nice browse back this forum here though its been while since surrounded myself with somewhat yes would say more than physical appearance certain degree what are they unsure they just generally cannot terrible person standards like lurk case least its mostly because tend believe have nothing interesting contribute the conversation why add anything logging out purposely lurk seems like think wait for him swing first before taking further action but would not encourage them take swing part wants fight him though would you say there complexity simplicity normally vote for whoever amuses the most perhaps one day will care more edit other than what amuses the time ill vote for whatever would apparently benefit the most long distance hard intj and what effects their sanity levels mental illness think about which thoughts wish express written format then proceed perform the physical movements necessary express the required thoughts whichever medium required got this seems right always score intj extra heavy the introversion like this like the sound the content lyrics are not too important the sound what value most favourite probably not heard others maybe although internally acutely aware any mistakes have made assuming they would listen would each give them solo task that leads the successful completion the project how would make them listen explain that the success the fan the idea that celestial being died and its essence became everything know what kind sacrifice would have made create such wondrous things stars planets cosmic nah since when you need manager present make returns was because work there check the company policy this why did they want save receipt not sure perhaps remember first encounter with this type music was boa hurricane venus saw played starcraft stream round something and was kind hooked not follow watch much what happens when this riddle gets solved what use this information haha clearly not fan riddles edit thoughts girlfriend asked few riddles and was infuriating for kept pressing them talk about something they did not want regret deeply not hands that summon desire want nothing with that puzzle box nope nope nope used think life was dull and everything was boring then realised that have not actually done everything felt incorrect thinking that way ahh yes the pain living its beautiful not got your score for openness was this the moderate range your score for conscientiousness was low your score for extraversion was low your score for what you mean humanities like think about the direction the human race species and like appreciate where have come from guess love humanities attempt think less about things more decisive and confident decisions thoughts c

```
annot help but play devils advocate with myself guess this ties into num
ber change ahh yes that sinking feeling when something goes incredibly w
rong not plan how get over embrace the madness how are you going deal wi
th your scenario now with all these new like imagine kind like this exce
pt the ring billion times higher and nearly impossible scale and then th
e middle beautiful garden and house where like chill you look the first
sentence most them you can gather the feeling they are trying portray fo
llowed rather nice scenario yes feel those feels but not for those think
its the introversion crossed with the feels that does for first encounte
red mbti hmm was personality test took highschool did not care even pay
attention until later stumbled upon somehow and was interested not too i
nto treat self question can intjs try hards answer sure not our values f
riendship are clearly different emotional investment any kind big deal h
uge deal have not appreciated and reciprocated soul destroying petty per
haps but why bother with someone who not going give what want when can j
ust move what waste time and emotional energy hopefully for you this fri
end yours'
```

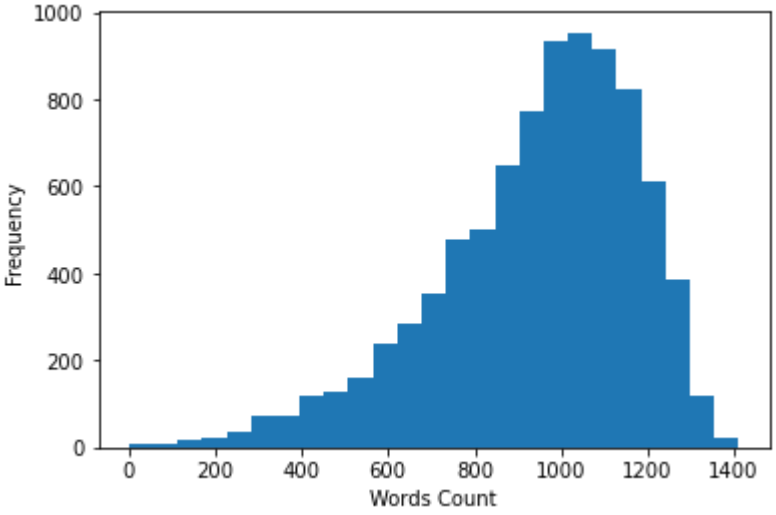
In [21]:
```
data["words_count"] = data["cleaned_post"].apply(lambda x: len(x.split()))
data.head(5)
```

Out[21]:

	type	posts	E-I	N-S	F-T	J-P	cleaned_post	words_cou
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	I	N	F	J	and intj moments sportscenter not top ten play...	4
1	ENTP	'I'm finding the lack of me in these posts ver...	E	N	T	P	finding the lack these posts very alarming sex...	8
2	INTP	'Good one _____ https://www.youtube.com/wat...	I	N	T	P	good one course which say know that blessing a...	6
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	I	N	T	J	dear intp enjoyed our conversation the other d...	8
4	ENTJ	'You're fired.|||That's another silly misconce...	E	N	T	J	you are fired that another silly misconception...	7

In [22]:
```
def plot_counts(df, column, xlabel):
    fig = plt.figure()
    plt.xlabel(xlabel)
    plt.ylabel("Frequency")
    df[column].plot.hist(bins=25)
```

In [23]:
```
plot_counts(data, column="words_count", xlabel="Words Count")
```

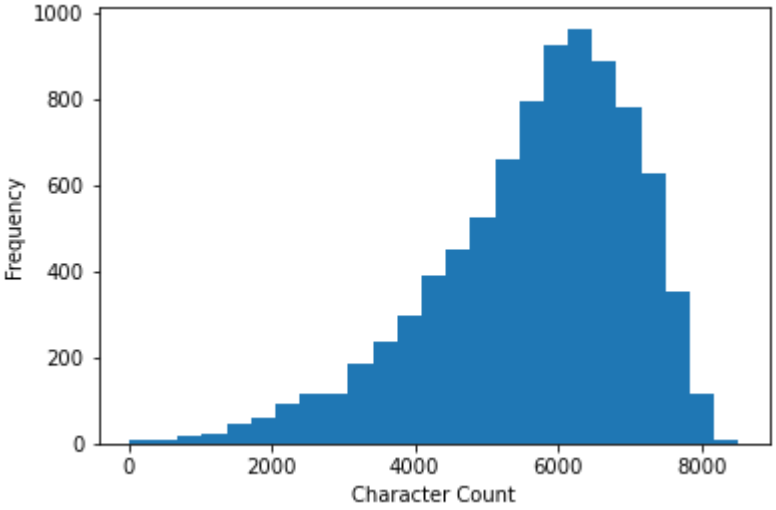


In [24]:
```
data["char_count"] = data["cleaned_post"].apply(lambda x: len(x))
data.head(5)
```

Out[24]:

	type	posts	E-I	N-S	F-T	J-P	cleaned_post	words_cou
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	I	N	F	J	and intj moments sportscenter not top ten play...	4
1	ENTP	'I'm finding the lack of me in these posts ver...	E	N	T	P	finding the lack these posts very alarming sex...	8
2	INTP	'Good one _____ https://www.youtube.com/wat...	I	N	T	P	good one course which say know that blessing a...	6
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	I	N	T	J	dear intp enjoyed our conversation the other d...	8
4	ENTJ	'You're fired.|||That's another silly misconce...	E	N	T	J	you are fired that another silly misconception...	7

In [25]:
```
plot_counts(data, column="char_count", xlabel="Character Count")
```



Most Frequent Words

In [26]:
```
stopword_list = stopwords.words("english")
```

In [27]:
```
def get_most_frequent(data, stop_words, column="cleaned_post", top=25):
    df = data[column].apply(lambda x: " ".join([word for word in x.split(
    counter = Counter(" ".join(df).split())
    return counter.most_common(top)
```

In [28]:
```
most_frequents = get_most_frequent(data, stopword_list)
most_frequents[:10]
```

Out[28]:
```
[('like', 69678),
 ('would', 52964),
 ('think', 49837),
 ('people', 48150),
 ('know', 38174),
 ('one', 37173),
 ('really', 35343),
 ('get', 30806),
 ('time', 27610),
 ('feel', 23337)]
```

In [29]:
```
def show_most_frequents(most_frequent_words, top=20):
    most_frequent_df = pd.DataFrame(most_frequent_words)
    plt.figure(figsize=(16,4))
    my_cmap = plt.get_cmap("viridis")
    plt.bar(x=most_frequent_df.iloc[:top, 0], height=most_frequent_df.ilo
    plt.xlabel("Words", size=17)
    plt.ylabel("Counts", size=17)
    plt.title("Most Frequent Words", size = 20)
    plt.show()
```

In [30]:
```
show_most_frequents(most_frequents)
```

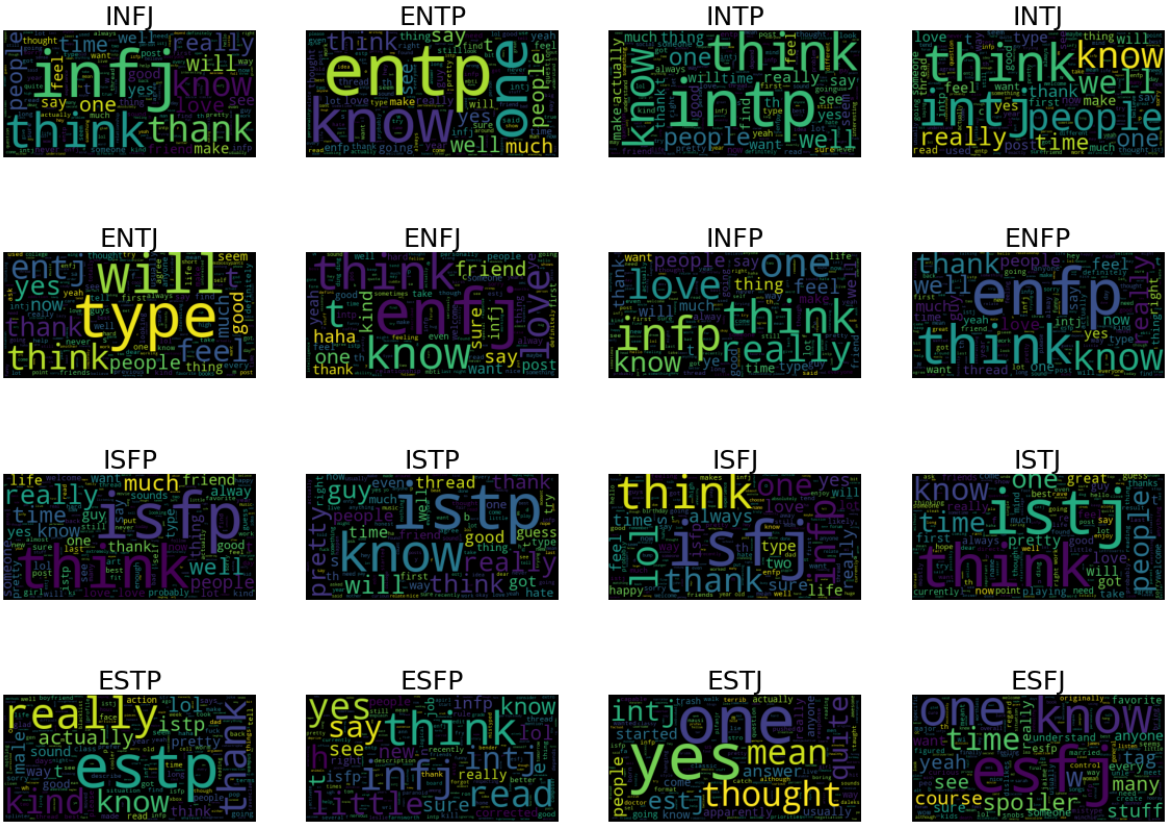
WordClouds

In [31]:
```
def show_wordcloud(data, stopword_list, column="cleaned_post"):
    fig = plt.figure(figsize=(15,5))
    wordcloud = WordCloud(background_color="black", min_font_size=5, stop
    plt.axis("off")
    plt.imshow(wordcloud)
    plt.show()
```

In [32]:
```
show_wordcloud(data, stopword_list)
```

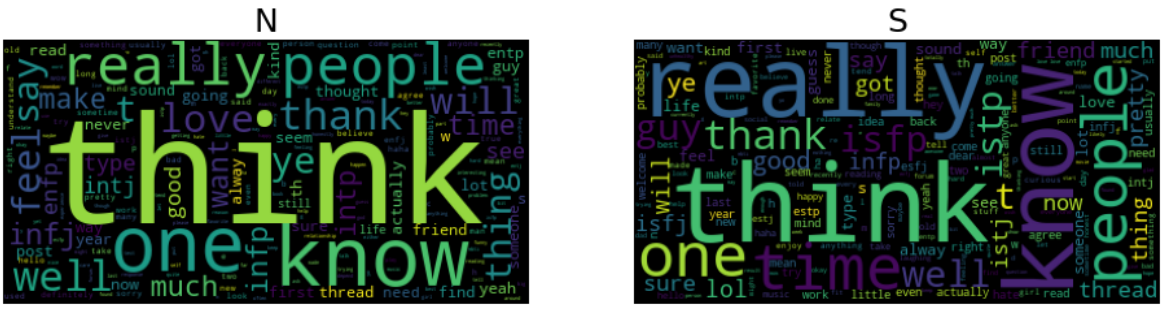
In [33]:
```
def show_sub_wordclouds(data, type_column, column, size, fig_size=(20,15)
    classes = data[type_column].unique()
    fig, ax = plt.subplots(len(classes), figsize=fig_size)
    j = 0
    for _class in classes:
        temp = data[data[type_column] == _class]
        wordcloud = WordCloud(background_color="black").generate(temp[col
        plt.subplot(*size, j+1)
        plt.title(_class, size=25)
        plt.imshow(wordcloud)
        plt.axis("off")
        j+=1
```

In [34]:
```
show_sub_wordclouds(data, type_column="type" , column="cleaned_post", siz
```



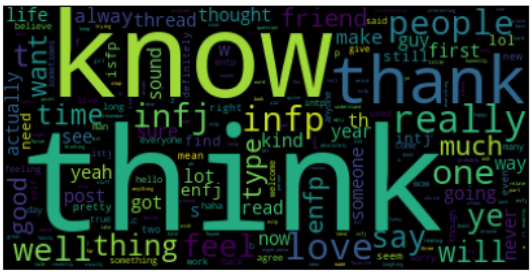
In [35]:
```
show_sub_wordclouds(data, type_column="E-I" , column="cleaned_post", size
```

In [36]:
```
show_sub_wordclouds(data, type_column="N-S" , column="cleaned_post", size
```

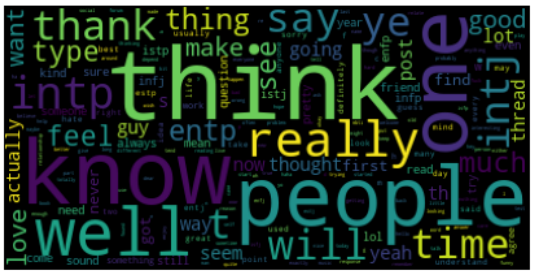


In [37]:
```
show_sub_wordclouds(data, type_column="F-T" , column="cleaned_post", size
```

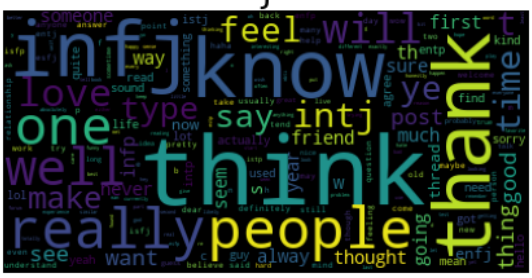
F



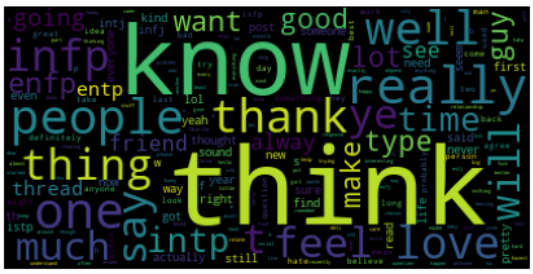
T



J



P



N-Grams

In [39]:
```
def get_ngrams(data, n_gram, new_column, column="cleaned_post"):
    data["tokenized"]   = data[column].apply(lambda x: x.split())
    data["sw_removal"] = data["tokenized"].apply(lambda x: [y for y in x
    data[new_column]    = data["sw_removal"].apply(lambda x: list(ngrams(x
    data.drop(columns  = ["tokenized", "sw_removal"], inplace=True)
    return data
```

In [40]:
```
data = get_ngrams(data, n_gram=2, new_column="bigrams")
data.head()
```

Out[40]:

	type	posts	E-I	N-S	F-T	J-P	cleaned_post	words_cou
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	I	N	F	J	and intj moments sportscenter not top ten play...	4
1	ENTP	'I'm finding the lack of me in these posts ver...	E	N	T	P	finding the lack these posts very alarming sex...	8
2	INTP	'Good one _____ https://www.youtube.com/wat...	I	N	T	P	good one course which say know that blessing a...	6
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	I	N	T	J	dear intp enjoyed our conversation the other d...	8
4	ENTJ	'You're fired.|||That's another silly misconce...	E	N	T	J	you are fired that another silly misconception...	7

In [41]:
```
data = get_ngrams(data, n_gram=3, new_column="trigrams")
data.head()
```

Out[41]:

	type	posts	E-I	N-S	F-T	J-P	cleaned_post	words_cou
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	I	N	F	J	and intj moments sportscenter not top ten play...	4
1	ENTP	'I'm finding the lack of me in these posts ver...	E	N	T	P	finding the lack these posts very alarming sex...	8
2	INTP	'Good one _____ https://www.youtube.com/wat...	I	N	T	P	good one course which say know that blessing a...	6
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	I	N	T	J	dear intp enjoyed our conversation the other d...	8
4	ENTJ	'You're fired.|||That's another silly misconce...	E	N	T	J	you are fired that another silly misconception...	7

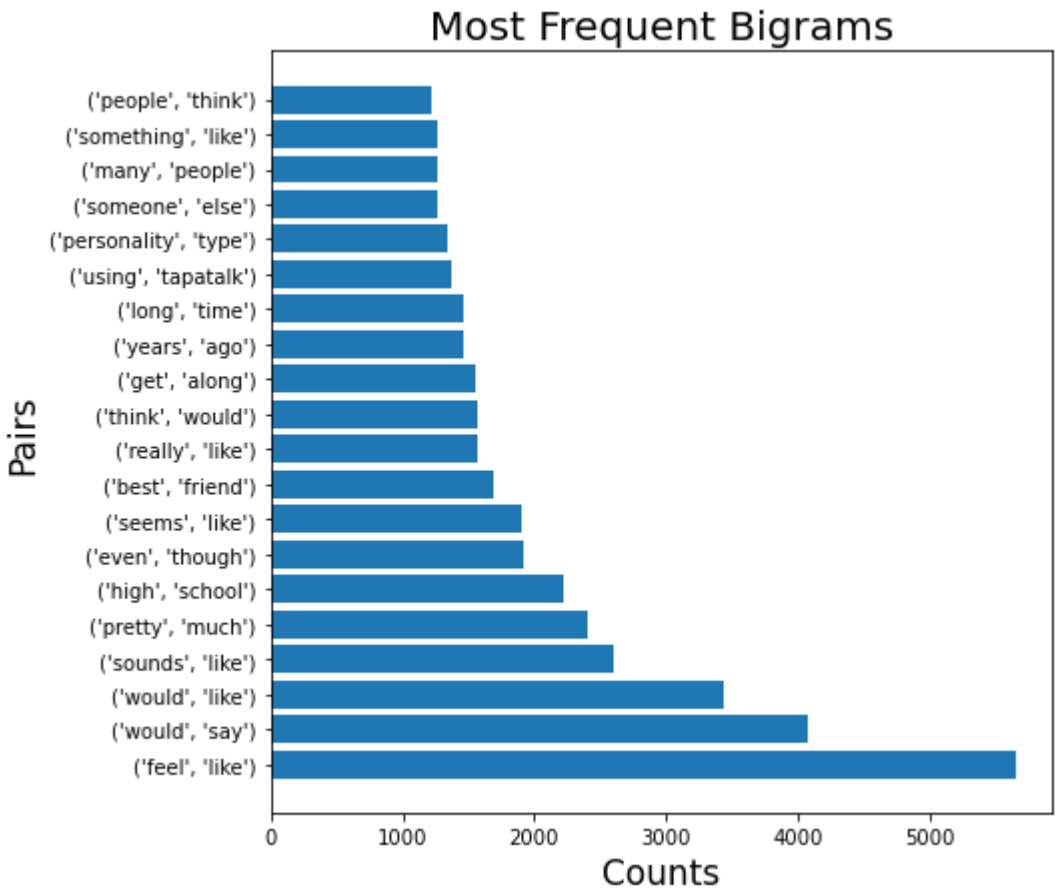
In [42]:
```
def most_common_ngram(data, column, top=20):
    temp = []
    for index, row in data.iterrows():
        temp += row[column]
    most_common = Counter(temp).most_common(top)
    return most_common
```

In [43]:
```
def plot_n_grams(ngrams, title, top=20):
    ngram_df = pd.DataFrame(ngrams)
    ngram_df.iloc[:, 0] = ngram_df.iloc[:,0].astype(str)
    plt.figure(figsize=(7,7))
    plt.barh(y=ngram_df.iloc[:top, 0], width=ngram_df.iloc[:top, 1])
    plt.xlabel("Counts", size=17)
    plt.ylabel("Pairs", size=17)
    plt.title(title, size = 20)
    plt.show()
```

In [44]:
```
bigrams_most_common = most_common_ngram(data, "bigrams")
bigrams_most_common
```

Out[44]:
```
[(('feel', 'like'), 5642),
 (('would', 'say'), 4073),
 (('would', 'like'), 3429),
 (('sounds', 'like'), 2606),
 (('pretty', 'much'), 2409),
 (('high', 'school'), 2216),
 (('even', 'though'), 1922),
 (('seems', 'like'), 1902),
 (('best', 'friend'), 1692),
 (('really', 'like'), 1576),
 (('think', 'would'), 1573),
 (('get', 'along'), 1551),
 (('years', 'ago'), 1460),
 (('long', 'time'), 1459),
 (('using', 'tapatalk'), 1376),
 (('personality', 'type'), 1337),
 (('someone', 'else'), 1272),
 (('many', 'people'), 1270),
 (('something', 'like'), 1267),
 (('people', 'think'), 1219)]
```

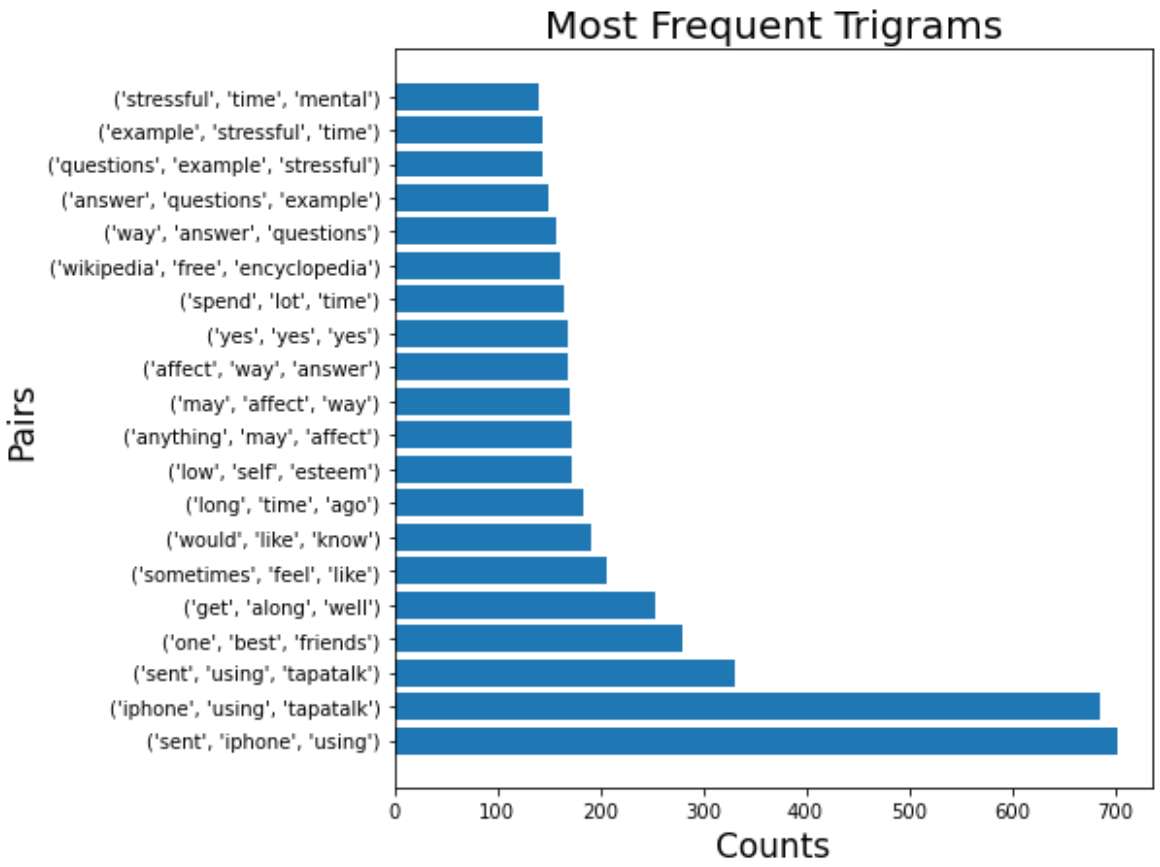
In [45]:
```
plot_n_grams(bigrams_most_common, title="Most Frequent Bigrams")
```



In [46]:
```
trigrams_most_common = most_common_ngram(data, "trigrams")
trigrams_most_common
```

Out[46]:
```
[(('sent', 'iphone', 'using'), 702),
 (('iphone', 'using', 'tapatalk'), 686),
 (('sent', 'using', 'tapatalk'), 331),
 (('one', 'best', 'friends'), 280),
 (('get', 'along', 'well'), 253),
 (('sometimes', 'feel', 'like'), 207),
 (('would', 'like', 'know'), 192),
 (('long', 'time', 'ago'), 183),
 (('low', 'self', 'esteem'), 173),
 (('anything', 'may', 'affect'), 173),
 (('may', 'affect', 'way'), 171),
 (('affect', 'way', 'answer'), 168),
 (('yes', 'yes', 'yes'), 168),
 (('spend', 'lot', 'time'), 165),
 (('wikipedia', 'free', 'encyclopedia'), 162),
 (('way', 'answer', 'questions'), 158),
 (('answer', 'questions', 'example'), 150),
 (('questions', 'example', 'stressful'), 145),
 (('example', 'stressful', 'time'), 144),
 (('stressful', 'time', 'mental'), 140)]
```

In [47]:
```
plot_n_grams(trigrams_most_common, title="Most Frequent Trigrams")
```



preprocessing

In [49]:
```
def remove_stopwords(data, stopword_list, column="cleaned_post"):
    data[column] = data[column].apply(word_tokenize)
    data[column] = data[column].apply(lambda x: [word for word in x if no
    return data
```

In [50]:
```
def apply_lemmatization(text):
    lemmatizer = WordNetLemmatizer()
```

```
    return [lemmatizer.lemmatize(w) for w in text]
```

In [51]:
```
def lemmatize(data, stopword_list, column="cleaned_post"):
    data[column] = data[column].apply(apply_lemmatization)
    data[column] = data[column].apply(" ".join)
    return data
```

In [52]:
```
data = remove_stopwords(data, stopword_list)
```

In [53]:
```
data = lemmatize(data, stopword_list)
```

In [54]:
```
data.head()
```

Out[54]:

	type	posts	E-I	N-S	F-T	J-P	cleaned_post	words_coun
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	I	N	F	J	intj moment sportscenter top ten play prank li...	45
1	ENTP	'I'm finding the lack of me in these posts ver...	E	N	T	P	finding lack post alarming sex boring position...	87
2	INTP	'Good one _____ https://www.youtube.com/wat...	I	N	T	P	good one course say know blessing curse absolu...	65
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	I	N	T	J	dear intp enjoyed conversation day esoteric ga...	82
4	ENTJ	'You're fired.|||That's another silly misconce...	E	N	T	J	fired another silly misconception approaching ...	78

In [55]:
```
training_data = data[["cleaned_post", "E-I", "N-S", "F-T", "J-P"]].copy()
training_data.head(5)
```

Out[55]:

	cleaned_post	E-I	N-S	F-T	J-P
0	intj moment sportscenter top ten play prank li...	I	N	F	J
1	finding lack post alarming sex boring position...	E	N	T	P
2	good one course say know blessing curse absolu...	I	N	T	P
3	dear intp enjoyed conversation day esoteric ga...	I	N	T	J
4	fired another silly misconception approaching ...	E	N	T	J

In [56]:
```
def make_dummies(data, columns=["E-I", "N-S", "F-T", "J-P"]):
    for column in columns:
        temp_dummy = pd.get_dummies(data[column], prefix="type")
        data = data.join(temp_dummy)
    return data
```

In [57]:
```
training_data = make_dummies(training_data)
training_data.head()
```

Out[57]:

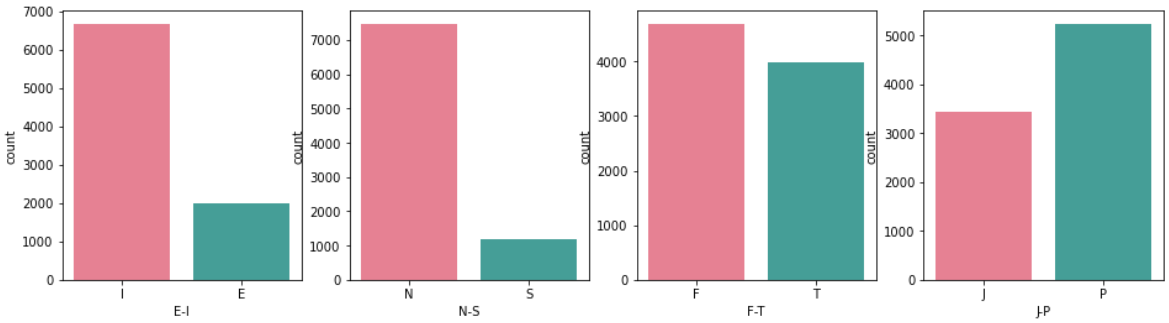
	cleaned_post	E-I	N-S	F-T	J-P	type_E	type_I	type_N	type_S	type_F	type_T
0	intj moment sportscenter top ten play prank li...	I	N	F	J	0	1	1	0	1	0
1	finding lack post alarming sex boring position...	E	N	T	P	1	0	1	0	0	1
2	good one course say know blessing curse absolu...	I	N	T	P	0	1	1	0	0	1
3	dear intp enjoyed conversation day esoteric ga...	I	N	T	J	0	1	1	0	0	1
4	fired another silly misconception approaching ...	E	N	T	J	1	0	1	0	0	1

Handling Imbalanced Data

In [58]:
```
X = training_data[["cleaned_post"]]
y = training_data.drop(columns=["cleaned_post"])
```

In [61]:
```
def show_distribution(data, x=["E-I", "N-S", "F-T","J-P"], fig_size=(16,4
    fig, ax = plt.subplots(len(x), figsize=fig_size)
    j = 0
    for _x in x:
        plt.subplot(1,4, j+1)
        sns.countplot(x=_x, data=data, palette=palette)
        plt.xticks(size=xticks_size)
        j+=1

show_distribution(data)
```



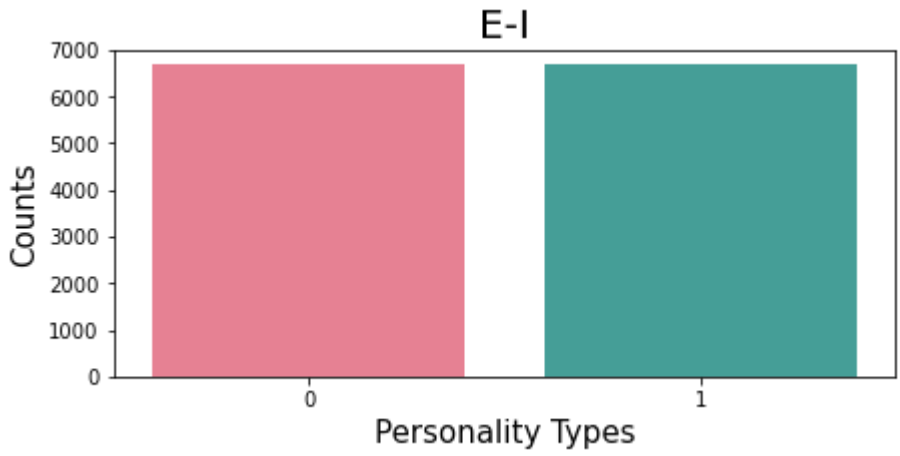
In [63]:
```
from imblearn.over_sampling import RandomOverSampler
```

In [64]:
```
oversample = RandomOverSampler()
```

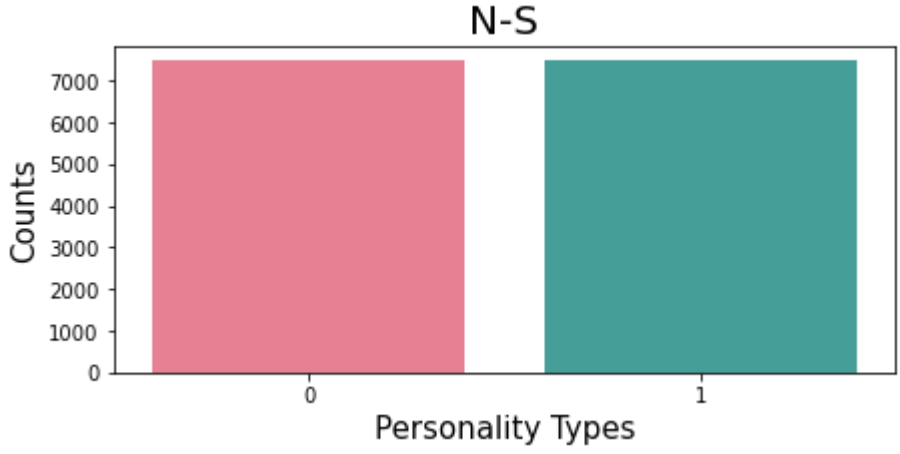
In [65]:
```
y_ei = y["type_E"]
y_ns = y["type_N"]
y_ft = y["type_F"]
y_jp = y["type_J"]
```

In [66]:
```
X_over_ei, y_over_ei = oversample.fit_resample(X, y_ei)
X_over_ns, y_over_ns = oversample.fit_resample(X, y_ns)
X_over_ft, y_over_ft = oversample.fit_resample(X, y_ft)
X_over_jp, y_over_jp = oversample.fit_resample(X, y_jp)
```

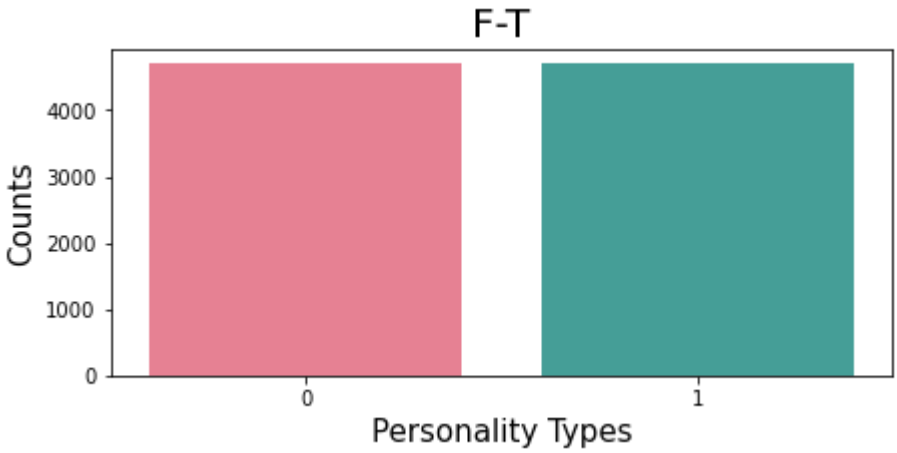
In [67]:
```
show_class_distribution(data=X_over_ei, x=y_over_ei, figsize=(7,3), title
```



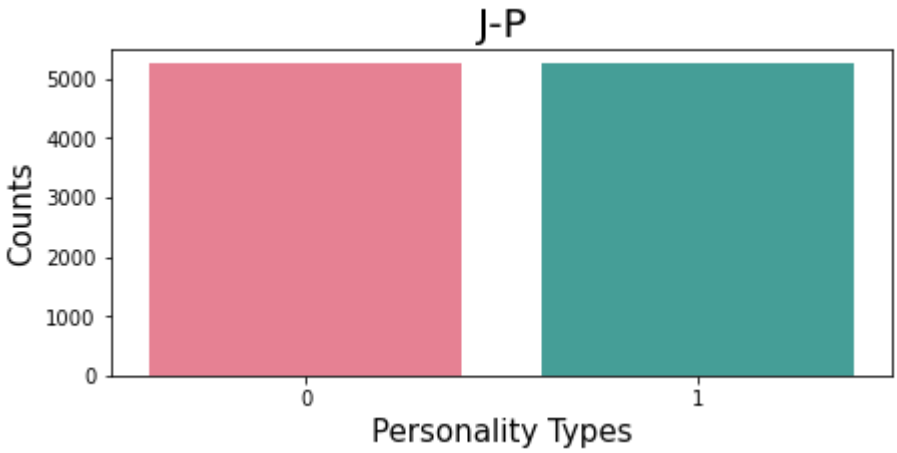
In [68]:
```
show_class_distribution(data=X_over_ns, x=y_over_ns, figsize=(7,3), title
```



In [69]:
```
show_class_distribution(data=X_over_ft, x=y_over_ft, figsize=(7,3), title
```



In [70]:
```
show_class_distribution(data=X_over_jp, x=y_over_jp, figsize=(7,3), title
```



Train-test split for each classes

In [71]:
```
from sklearn.model_selection import train_test_split
```

In [72]:
```
X_train_ei, X_test_ei, y_train_ei, y_test_ei = train_test_split(X_over_ei
X_train_ns, X_test_ns, y_train_ns, y_test_ns = train_test_split(X_over_ns
X_train_ft, X_test_ft, y_train_ft, y_test_ft = train_test_split(X_over_ft
X_train_jp, X_test_jp, y_train_jp, y_test_jp = train_test_split(X_over_jp
```

In [73]:
```
X_train_ei = X_train_ei['cleaned_post']
X_train_ns = X_train_ns['cleaned_post']
X_train_ft = X_train_ft['cleaned_post']
X_train_jp = X_train_jp['cleaned_post']
```

In [74]:
```
X_test_ei = X_test_ei['cleaned_post']
X_test_ns = X_test_ns['cleaned_post']
X_test_ft = X_test_ft['cleaned_post']
X_test_jp = X_test_jp['cleaned_post']
```

In [75]:
```
y_train_ei.name, y_test_ei.name = "E-I", "E-I"
y_train_ns.name, y_test_ns.name = "N-S", "N-S"
y_train_ft.name, y_test_ft.name = "F-T", "F-T"
y_train_jp.name, y_test_jp.name = "J-P", "J-P"
```

In [76]:
```
y_all_train = [y_train_ei, y_train_ns, y_train_ft,  y_train_jp]
y_all_test  = [y_test_ei, y_test_ns, y_test_ft, y_test_jp]
```

TF-IDF Vectorizer

In [77]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [78]:
```
vectorizer = TfidfVectorizer(max_features=10000)
```

In [79]:
```
vectorizer.fit(X_train_ei)
```

Out[79]:
```
TfidfVectorizer(max_features=10000)
```

In [80]:
```
X_train_ei = vectorizer.transform(X_train_ei)
X_test_ei  = vectorizer.transform(X_test_ei)

X_train_ns = vectorizer.transform(X_train_ns)
X_test_ns  = vectorizer.transform(X_test_ns)

X_train_ft = vectorizer.transform(X_train_ft)
X_test_ft  = vectorizer.transform(X_test_ft)

X_train_jp = vectorizer.transform(X_train_jp)
X_test_jp  = vectorizer.transform(X_test_jp)
```

In [82]:
```
x_all_train = [X_train_ei, X_train_ns, X_train_ft, X_train_jp]
x_all_test  = [X_test_ei, X_test_ns, X_test_ft, X_test_jp]
```

In [83]:
```
tf_idf = pd.DataFrame(X_test_ei.toarray(), columns=vectorizer.get_feature
tf_idf.head(10)
```

Out[83]:

	aang	ab	aback	abandon	abandoned	abandonment	abbey	abbreviation	ab
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

10 rows × 10000 columns

Model Creation & Model Training & Model Saving

In [84]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import xgboost
import pickle
from sklearn import metrics
```

In [85]:
```
def create_models():
    nb_clf  = MultinomialNB(alpha=0.01)
    svm_clf = SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
    dt_clf  = DecisionTreeClassifier(max_depth=7)
    rf_clf  = RandomForestClassifier(n_estimators=750)
    xgb_clf = xgboost.XGBClassifier(use_label_encoder=False)
    return {"NaiveBayes":nb_clf, "SVM":svm_clf, "DecisionTree":dt_clf, "R
```

Model Performance Evaluation with accuracy & f1-score & roc-auc score

In [88]:
```
_metrics = ["Accuracy", "Accuracy", "Accuracy" ,"Accuracy", "Precision",
_types   = ["E-I", "N-S", "F-T", "J-P", "E-I", "N-S", "F-T", "J-P", "E-I"
_columns = ["NaiveBayes", "SVM", "DecisionTree", "RandomForest", "Xgboost
```

In [89]:
```
evaluation_df = pd.DataFrame(columns=_columns, index=[_metrics, _types])
evaluation_df
```

Out[89]:

		NaiveBayes	SVM	DecisionTree	RandomForest	Xgboost
Accuracy	**E-I**	NaN	NaN	NaN	NaN	NaN
	N-S	NaN	NaN	NaN	NaN	NaN
	F-T	NaN	NaN	NaN	NaN	NaN
	J-P	NaN	NaN	NaN	NaN	NaN
Precision	**E-I**	NaN	NaN	NaN	NaN	NaN
	N-S	NaN	NaN	NaN	NaN	NaN
	F-T	NaN	NaN	NaN	NaN	NaN
	J-P	NaN	NaN	NaN	NaN	NaN
Recall	**E-I**	NaN	NaN	NaN	NaN	NaN
	N-S	NaN	NaN	NaN	NaN	NaN
	F-T	NaN	NaN	NaN	NaN	NaN
	J-P	NaN	NaN	NaN	NaN	NaN
F1-Score	**E-I**	NaN	NaN	NaN	NaN	NaN
	N-S	NaN	NaN	NaN	NaN	NaN
	F-T	NaN	NaN	NaN	NaN	NaN
	J-P	NaN	NaN	NaN	NaN	NaN
Roc-Auc Score	**E-I**	NaN	NaN	NaN	NaN	NaN
	N-S	NaN	NaN	NaN	NaN	NaN
	F-T	NaN	NaN	NaN	NaN	NaN
	J-P	NaN	NaN	NaN	NaN	NaN

In [90]:
```
models = create_models()
models
```

Out[90]:
```
{'NaiveBayes': MultinomialNB(alpha=0.01),
 'SVM': SVC(gamma='auto', kernel='linear'),
 'DecisionTree': DecisionTreeClassifier(max_depth=7),
 'RandomForest': RandomForestClassifier(n_estimators=750),
 'Xgboost': XGBClassifier(base_score=None, booster=None, callbacks=None,
               colsample_bylevel=None, colsample_bynode=None,
               colsample_bytree=None, early_stopping_rounds=None,
               enable_categorical=False, eval_metric=None, gamma=None,
               gpu_id=None, grow_policy=None, importance_type=None,
               interaction_constraints=None, learning_rate=None, max_bin
=None,
               max_cat_to_onehot=None, max_delta_step=None, max_depth=No
ne,
               max_leaves=None, min_child_weight=None, missing=nan,
               monotone_constraints=None, n_estimators=100, n_jobs=None,
               num_parallel_tree=None, predictor=None, random_state=Non
e,
               reg_alpha=None, reg_lambda=None, ...)}
```

In [91]:
```
for model_item in models.items():
    for X_train, X_test, y_train, y_test in zip(x_all_train, x_all_test,
        # Model creation and prediction
        model = model_item[1]
        print(f"{model} is training for {y_train.name}...")
        model.fit(X_train, y_train)
        pred = model.predict(X_test)
        # Performance evaluation metrics
        evaluation_df.loc["Accuracy",y_train.name][model_item[0]]      =
        evaluation_df.loc["Precision",y_train.name][model_item[0]]     =
        evaluation_df.loc["Recall",y_train.name][model_item[0]]        =
        evaluation_df.loc["F1-Score",y_train.name][model_item[0]]      =
        evaluation_df.loc["Roc-Auc Score",y_train.name][model_item[0]] =
        # Save model
        filename = f'saved-models/{model_item[0]}_{y_test.name}.sav'
        pickle.dump(model,open(filename, 'wb'))
```

```
MultinomialNB(alpha=0.01) is training for E-I...
MultinomialNB(alpha=0.01) is training for N-S...
MultinomialNB(alpha=0.01) is training for F-T...
MultinomialNB(alpha=0.01) is training for J-P...
SVC(gamma='auto', kernel='linear') is training for E-I...
SVC(gamma='auto', kernel='linear') is training for N-S...
SVC(gamma='auto', kernel='linear') is training for F-T...
SVC(gamma='auto', kernel='linear') is training for J-P...
DecisionTreeClassifier(max_depth=7) is training for E-I...
DecisionTreeClassifier(max_depth=7) is training for N-S...
DecisionTreeClassifier(max_depth=7) is training for F-T...
DecisionTreeClassifier(max_depth=7) is training for J-P...
RandomForestClassifier(n_estimators=750) is training for E-I...
RandomForestClassifier(n_estimators=750) is training for N-S...
RandomForestClassifier(n_estimators=750) is training for F-T...
RandomForestClassifier(n_estimators=750) is training for J-P...
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, gamma=None,
              gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=No
ne,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, n_estimators=100, n_jobs=None,
              num_parallel_tree=None, predictor=None, random_state=None,
              reg_alpha=None, reg_lambda=None, ...) is training for E-I...
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwis
e',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weigh
t=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_stat
e=0,
              reg_alpha=0, reg_lambda=1, ...) is training for N-S...
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwis
e',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weigh
t=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_stat
e=0,
              reg_alpha=0, reg_lambda=1, ...) is training for F-T...
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwis
e',
              importance_type=None, interaction_constraints='',
```

```
                learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
                max_delta_step=0, max_depth=6, max_leaves=0, min_child_weigh
t=1,
                missing=nan, monotone_constraints='()', n_estimators=100,
                n_jobs=0, num_parallel_tree=1, predictor='auto', random_stat
e=0,
                reg_alpha=0, reg_lambda=1, ...) is training for J-P...
```

In [93]:
```
evaluation_df
```

Out[93]:

		NaiveBayes	SVM	DecisionTree	RandomForest	Xgboost
Accuracy	**E-I**	0.823	0.896	0.796	0.952	0.936
	N-S	0.905	0.953	0.797	0.993	0.972
	F-T	0.812	0.856	0.749	0.845	0.845
	J-P	0.732	0.81	0.735	0.84	0.841
Precision	**E-I**	0.827	0.882	0.835	0.986	0.914
	N-S	0.908	0.98	0.765	0.991	0.998
	F-T	0.806	0.851	0.747	0.831	0.843
	J-P	0.728	0.801	0.772	0.91	0.827
Recall	**E-I**	0.818	0.915	0.74	0.918	0.964
	N-S	0.903	0.926	0.863	0.995	0.947
	F-T	0.82	0.861	0.751	0.866	0.846
	J-P	0.735	0.821	0.662	0.752	0.859
F1-Score	**E-I**	0.823	0.898	0.785	0.951	0.938
	N-S	0.905	0.953	0.811	0.993	0.972
	F-T	0.813	0.856	0.749	0.848	0.845
	J-P	0.731	0.811	0.713	0.824	0.843
Roc-Auc Score	**E-I**	0.823	0.896	0.797	0.952	0.936
	N-S	0.905	0.954	0.797	0.993	0.973
	F-T	0.812	0.856	0.749	0.845	0.845
	J-P	0.732	0.81	0.734	0.84	0.841

In [ ]:
```
### Save Tf-Idf Vectorizer
```

In [95]:
```
filename = 'vectorizer.pkl'
pickle.dump(vectorizer,open(filename, 'wb'))
```

In [ ]:
```

```