In [58]:
```
pip install stop-words
```

```
Note: you may need to restart the kernel to use updated packages.Collecting stop-words

  Downloading stop-words-2018.7.23.tar.gz (31 kB)
Building wheels for collected packages: stop-words
  Building wheel for stop-words (setup.py): started
  Building wheel for stop-words (setup.py): finished with status 'done'
  Created wheel for stop-words: filename=stop_words-2018.7.23-py3-none-any.whl size=32911 sha256=dbce13725f8a32
a778f2b7e084eefc48e4c8c13316f04c035d2a15b6d92cd4e3
  Stored in directory: c:\users\ragavb\appdata\local\pip\cache\wheels\da\d8\66\395317506a23a9d1d7de433ad6a7d9e6
e16aab48cf028a0f60
Successfully built stop-words
Installing collected packages: stop-words
Successfully installed stop-words-2018.7.23
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
df=pd.read_excel(r'D:\Malignant-Comments-Classifier-Project\Malignant Comments Classifier Project\train.xlsx')
df
```

Out[3]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...
159566	ffe987279560d7ff	":::::And for the second time of asking, when ...	0	0	0	0	0	0
159567	ffea4adeee384e90	You should be ashamed of yourself \n\nThat is ...	0	0	0	0	0	0
159568	ffee36eab5c267c9	Spitzer \n\nUmm, theres no actual article for ...	0	0	0	0	0	0
159569	fff125370e4aaaf3	And it looks like it was actually you who put ...	0	0	0	0	0	0
159570	fff46fc426af1f9a	"\nAnd ... I really don't think you understand...	0	0	0	0	0	0

159571 rows × 8 columns

In [4]:
```
df.dtypes
```

Out[4]:
```
id                  object
comment_text        object
malignant            int64
highly_malignant     int64
rude                 int64
threat               int64
abuse                int64
loathe               int64
dtype: object
```

In [7]:
```
df.shape
```

Out[7]:
```
(159571, 8)
```

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

Out[8]:

	malignant	highly_malignant	rude	threat	abuse	loathe
count	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000
mean	0.095844	0.009996	0.052948	0.002996	0.049364	0.008805
std	0.294379	0.099477	0.223931	0.054650	0.216627	0.093420
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [9]:
```
df.isnull()
```

Out[9]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
159566	False	False	False	False	False	False	False	False
159567	False	False	False	False	False	False	False	False
159568	False	False	False	False	False	False	False	False
159569	False	False	False	False	False	False	False	False
159570	False	False	False	False	False	False	False	False

159571 rows × 8 columns

In [4]:

```
df=df.reindex(np.random.permutation(df.index))
```

In [11]:

```
comment =df['comment_text']
print(comment.head())
comment = comment.to_numpy()
```

```
52159     Bracket Bot Is Evil \nJonesy, please don't com...
128230    "\n\nRE:Illa J and Frank Nitt\nI was thinking ...
85872             The Change You Made Was Very Interesting.
2771      The very long discussion over st WP:SCN projec...
799       Content subsumed into Maneesh page (same entry...
Name: comment_text, dtype: object
```

In [49]:

```
df.astype({'comment_text':'string'}).dtypes
```

Out[49]:

```
id                  object
comment_text        string
malignant            int64
highly_malignant     int64
rude                 int64
threat               int64
abuse                int64
loathe               int64
dtype: object
```

In [12]:

```
label=df[['malignant', 'highly_malignant', 'rude', 'threat', 'abuse', 'loathe']]
print(label.head())
label=label.to_numpy()
```

```
   malignant  highly_malignant  rude  threat  abuse  loathe
0          0                 0     0       0      0       0
1          0                 0     0       0      0       0
2          0                 0     0       0      0       0
3          0                 0     0       0      0       0
4          0                 0     0       0      0       0
```

In [13]:

```
ct1,ct2 = 0,0
for i in range(label.shape[0]):
    ct=np.count_nonzero(label[i])
    if ct :
        ct1=ct1+1
    if ct>1:
        ct2=ct2+1
print(ct1)
print(ct2)
```

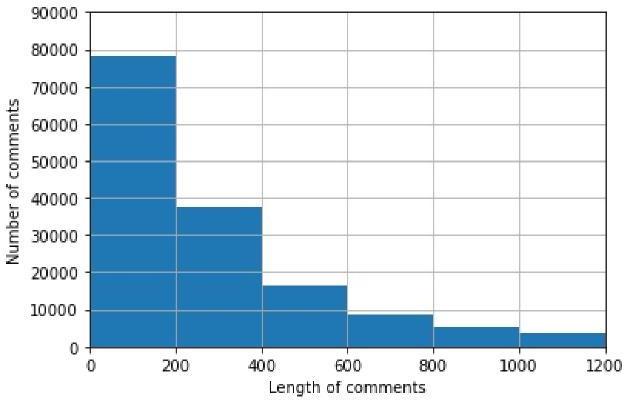
```
16225
9865
```

In [51]:

```
x = [len(comment[i]) for i in range(comment.shape[0])]

print('average length of comment: {:.3f}'.format(sum(x)/len(x)) )
bins = [1,200,400,600,800,1000,1200]
plt.hist(x, bins=bins)
plt.xlabel('Length of comments')
plt.ylabel('Number of comments')
plt.axis([0, 1200, 0, 90000])
plt.grid(True)
plt.show()
```

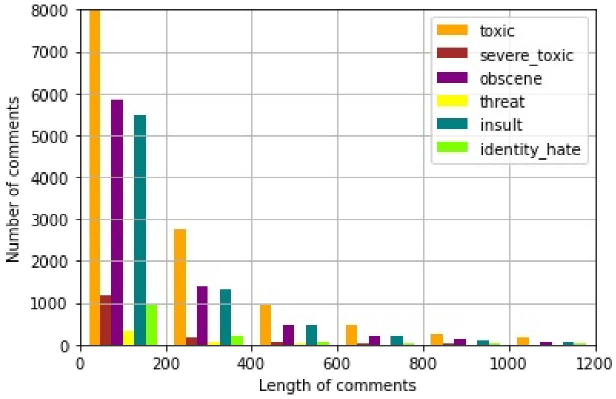
```
average length of comment: 394.139
```



In [52]:

```
y = np.zeros(label.shape)
for ix in range(comment.shape[0]):
    l = len(comment[ix])
    if label[ix][0] :
        y[ix][0] = l
    if label[ix][1] :
        y[ix][1] = l
    if label[ix][2] :
        y[ix][2] = l
    if label[ix][3] :
        y[ix][3] = l
    if label[ix][4] :
        y[ix][4] = l
    if label[ix][5] :
        y[ix][5] = l

labelsplt = ['toxic','severe_toxic','obscene','threat','insult','identity_hate']
color = ['orange','brown','purple','yellow','teal','chartreuse']
plt.hist(y,bins = bins,label = labelsplt,color = color)
plt.axis([0, 1200, 0, 8000])
plt.xlabel('Length of comments')
plt.ylabel('Number of comments')
plt.legend()
plt.grid(True)
plt.show()
```



In [53]:

```
comments = []
labels = []

for ix in range(comment.shape[0]):
    if len(comment[ix])<=400:
        comments.append(comment[ix])
        labels.append(label[ix])
```

In [54]:

```
labels = np.asarray(labels)
```

In [55]:

```
print(len(comments))
```

```
115893
```

In [56]:

```
import string
print(string.punctuation)
punctuation_edit = string.punctuation.replace('\'','') +"0123456789"
print (punctuation_edit)
outtab = "                                         "
trantab = str.maketrans(punctuation_edit, outtab)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
!"#$%&()*+,-./:;<=>?@[\]^_`{|}~0123456789
```

In [59]:

```python
from stop_words import get_stop_words
stop_words = get_stop_words('english')
stop_words.append('')

for x in range(ord('b'), ord('z')+1):
    stop_words.append(chr(x))
```

In [60]:

```python
print (stop_words)
```

```
['a', 'about', 'above', 'after', 'again', 'against', 'all', 'am', 'an', 'and', 'any', 'are', "aren't", 'as', 'a
t', 'be', 'because', 'been', 'before', 'being', 'below', 'between', 'both', 'but', 'by', "can't", 'cannot', 'co
uld', "couldn't", 'did', "didn't", 'do', 'does', "doesn't", 'doing', "don't", 'down', 'during', 'each', 'few',
'for', 'from', 'further', 'had', "hadn't", 'has', "hasn't", 'have', "haven't", 'having', 'he', "he'd", "he'll",
"he's", 'her', 'here', "here's", 'hers', 'herself', 'him', 'himself', 'his', 'how', "how's", 'i', "i'd", "i'll"
, "i'm", "i've", 'if', 'in', 'into', 'is', "isn't", 'it', "it's", 'its', 'itself', "let's", 'me', 'more', 'most
', "mustn't", 'my', 'myself', 'no', 'nor', 'not', 'of', 'off', 'on', 'once', 'only', 'or', 'other', 'ought', 'o
ur', 'ours', 'ourselves', 'out', 'over', 'own', 'same', "shan't", 'she', "she'd", "she'll", "she's", 'should',
"shouldn't", 'so', 'some', 'such', 'than', 'that', "that's", 'the', 'their', 'theirs', 'them', 'themselves', 't
hen', 'there', "there's", 'these', 'they', "they'd", "they'll", "they're", "they've", 'this', 'those', 'through
', 'to', 'too', 'under', 'until', 'up', 'very', 'was', "wasn't", 'we', "we'd", "we'll", "we're", "we've", 'were
', "weren't", 'what', "what's", 'when', "when's", 'where', "where's", 'which', 'while', 'who', "who's", 'whom',
'why', "why's", 'with', "won't", 'would', "wouldn't", 'you', "you'd", "you'll", "you're", "you've", 'your', 'yo
urs', 'yourself', 'yourselves', '', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
```

In [ ]:

```python
import nltk
nltk.download()
from nltk.stem import PorterStemmer, WordNetLemmatizer
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

In [ ]:

```python
lemmatiser = WordNetLemmatizer()
stemmer = PorterStemmer()

nltk.download('wordnet')
```

In [ ]:

```python
for i in range(len(comments)):
    comments[i] = comments[i].lower().translate(trantab)
    l = []
    for word in comments[i].split():
        l.append(stemmer.stem(lemmatiser.lemmatize(word,pos="v")))
    comments[i] = " ".join(l)
```

In [ ]:

```python
a = np.zeros((115893, 97680), dtype='int64')
a.nbytes
```

In [ ]:

```python
from sklearn.feature_extraction.text import CountVectorizer

#create object supplying our custom stop words
count_vector = CountVectorizer(stop_words=stop_words)
#fitting it to converts comments into bag of words format
tf = count_vector.fit_transform(comments).toarray()
```

In [ ]:

```python
print(tf.shape)
```

In [13]:

```python
def shuffle(matrix, target, test_proportion):
    ratio = int(matrix.shape[0]/test_proportion)
    X_train = matrix[ratio:,:]
    X_test =  matrix[:ratio,:]
    Y_train = target[ratio:,:]
    Y_test =  target[:ratio,:]
    return X_train, X_test, Y_train, Y_test

X_train, X_test, Y_train, Y_test = shuffle(label,3)

print(X_test.shape)
print(X_train.shape)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [13], in <cell line: 9>()
      6     Y_test =  target[:ratio,:]
      7     return X_train, X_test, Y_train, Y_test
----> 9 X_train, X_test, Y_train, Y_test = shuffle(label,3)
     11 print(X_test.shape)
     12 print(X_train.shape)

TypeError: shuffle() missing 1 required positional argument: 'test_proportion'
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js