In [1]:

```
from google.colab import files
files.upload()
# check to see if the file is there
!ls -lha kaggle.json
# install Kaggle API
!pip install kaggle --upgrade
# move file
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
# change permissions
!chmod 600 ~/.kaggle/kaggle.json
# download dataset
!kaggle competitions download -c jigsaw-toxic-comment-classification-challenge
```

Choose File **No file selected**

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle.json
-rw-r--r-- 1 root root 64 Jan 27 18:26 kaggle.json
Requirement already up-to-date: kaggle in /usr/local/lib/python3.6/dist-packages (1.5.10)
Requirement already satisfied, skipping upgrade: python-dateutil in /usr/local/lib/python
3.6/dist-packages (from kaggle) (2.8.1)
Requirement already satisfied, skipping upgrade: requests in /usr/local/lib/python3.6/dis
t-packages (from kaggle) (2.23.0)
Requirement already satisfied, skipping upgrade: python-slugify in /usr/local/lib/python3
.6/dist-packages (from kaggle) (4.0.1)
Requirement already satisfied, skipping upgrade: urllib3 in /usr/local/lib/python3.6/dist
-packages (from kaggle) (1.24.3)
Requirement already satisfied, skipping upgrade: tqdm in /usr/local/lib/python3.6/dist-pa
ckages (from kaggle) (4.41.1)
Requirement already satisfied, skipping upgrade: six>=1.10 in /usr/local/lib/python3.6/di
st-packages (from kaggle) (1.15.0)
Requirement already satisfied, skipping upgrade: certifi in /usr/local/lib/python3.6/dist
-packages (from kaggle) (2020.12.5)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in /usr/local/lib/python3.6
/dist-packages (from requests->kaggle) (2.10)
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in /usr/local/lib/pyth
on3.6/dist-packages (from requests->kaggle) (3.0.4)
Requirement already satisfied, skipping upgrade: text-unidecode>=1.3 in /usr/local/lib/py
thon3.6/dist-packages (from python-slugify->kaggle) (1.3)
Warning: Looks like you're using an outdated API Version, please consider updating (serve
r 1.5.10 / client 1.5.4)
Downloading sample_submission.csv.zip to /content
  0% 0.00/1.39M [00:00<?, ?B/s]
100% 1.39M/1.39M [00:00<00:00, 92.9MB/s]
Downloading test_labels.csv.zip to /content
  0% 0.00/1.46M [00:00<?, ?B/s]
100% 1.46M/1.46M [00:00<00:00, 208MB/s]
Downloading test.csv.zip to /content
 85% 20.0M/23.4M [00:00<00:00, 204MB/s]
100% 23.4M/23.4M [00:00<00:00, 216MB/s]
Downloading train.csv.zip to /content
 68% 18.0M/26.3M [00:00<00:00, 188MB/s]
100% 26.3M/26.3M [00:00<00:00, 168MB/s]
```

In [2]:

```
#unzip dataset
!unzip train.csv.zip
!unzip test.csv.zip
!unzip test_labels.csv.zip
```

```
Archive:  train.csv.zip
  inflating: train.csv
Archive:  test.csv.zip
```

```
  inflating: test.csv
Archive:  test_labels.csv.zip
  inflating: test_labels.csv
```

In [3]:

```
classes = ["toxic", "severe_toxic", "obscene", "threat", "insult", "identity_hate"]
```

In [4]:

```
import pandas as pd
import numpy as np
training= pd.read_csv('train.csv', sep=',')
validation = pd.read_csv('test.csv', sep = ',')
test_labels = pd.read_csv('test_labels.csv', sep = ',')
```

In [5]:

```
training.head(10)
```

Out[5]:

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
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
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

In [6]:

```
validation.head(10)
```

Out[6]:

	id	comment_text
0	00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...
1	0000247867823ef7	== From RfC == \n\n The title is fine as it is...
2	00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
3	00017563c3f7919a	:If you have a look back at the source, the in...
4	00017695ad8997eb	I don't anonymously edit articles at all.
5	0001ea8717f6de06	Thank you for understanding. I think very high...
6	00024115d4cbde0f	Please do not add nonsense to Wikipedia. Such ...
7	000247e83dcc1211	:Dear god this site is horrible.

	id	comment_text
8	00025358d4737918	" \n Only a fool can believe in such numbers. ...
9	00026d1092fe71cc	== Double Redirects == \n\n When fixing double...

In [7]:

```
test_labels.head(10)
```

Out[7]:

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	-1	-1	-1	-1	-1	-1
1	0000247867823ef7	-1	-1	-1	-1	-1	-1
2	00013b17ad220c46	-1	-1	-1	-1	-1	-1
3	00017563c3f7919a	-1	-1	-1	-1	-1	-1
4	00017695ad8997eb	-1	-1	-1	-1	-1	-1
5	0001ea8717f6de06	0	0	0	0	0	0
6	00024115d4cbde0f	-1	-1	-1	-1	-1	-1
7	000247e83dcc1211	0	0	0	0	0	0
8	00025358d4737918	-1	-1	-1	-1	-1	-1
9	00026d1092fe71cc	-1	-1	-1	-1	-1	-1

In [8]:

```
import re

def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)

    return input_txt

def preprocess(x, y):
  x[y] = np.vectorize(remove_pattern)(x[y], "@[\w]*")
  x[y] = x[y].str.replace("[^a-zA-Z#]", " ")
  x[y] = x[y].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))

preprocess(training, 'comment_text')
preprocess(validation, 'comment_text')
```

In [9]:

```
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
stop_words.update(['zero','one','two','three','four','five','six','seven','eight','nine'
,'ten','may','also','across','among','beside','however','yet','within', 'think', 'page']
)
re_stop_words = re.compile(r"\b(" + "|".join(stop_words) + ")\\W", re.I)
def removeStopWords(sentence):
    global re_stop_words
    return re_stop_words.sub(" ", sentence)

training['comment_text'] = training['comment_text'].apply(removeStopWords)
validation['comment_text'] = validation['comment_text'].apply(removeStopWords)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

In [10]:

```
training.head(10)
```

Out[10]:

Out[10]:

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation edits made username Hardcore Meta...	0	0	0	0	0	0
1	000103f0d9cfb60f	matches background colour seemingly stuck Th...	0	0	0	0	0	0
2	000113f07ec002fd	really trying edit constantly removing rele...	0	0	0	0	0	0
3	0001b41b1c6bb37e	make real suggestions improvement wondered se...	0	0	0	0	0	0
4	0001d958c54c6e35	hero chance remember that	0	0	0	0	0	0
5	00025465d4725e87	Congratulations well tools well talk	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER PISS AROUND WORK	1	1	1	0	1	0
7	00031b1e95af7921	vandalism Matt Shirvington article reverted ...	0	0	0	0	0	0
8	00037261f536c51d	Sorry word nonsense offensive Anyway intending...	0	0	0	0	0	0
9	00040093b2687caa	alignment subject contrary DuLithgow	0	0	0	0	0	0

In [11]:

```
training_labels = training[classes].values
print(training_labels.shape)
```

```
(159571, 6)
```

In [12]:

```
# do not run more than once!
test_labels = test_labels[classes].replace(0, 1)
test_labels = test_labels[classes].replace(-1, 0)
```

In [13]:

```
testlabels = test_labels[classes].values
print(testlabels.shape)
```

```
(153164, 6)
```

In [14]:

```
#combine them for analysis via wordcloud, I did this multiple times by accident which is why it is 4 times as wide
validation = pd.concat([validation, test_labels], axis=1)
validation.head(10)
```

Out[14]:

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	bitch Rule succesful ever whats hating mofu...	0	0	0	0	0	0
1	0000247867823ef7	title fine	0	0	0	0	0	0
2	00013b17ad220c46	Sources Zawe Ashton Lapland	0	0	0	0	0	0
3	00017563c3f7919a	look back source information updated correct ...	0	0	0	0	0	0
4	00017695ad8997eb	anonymously edit articles	0	0	0	0	0	0
5	0001ea8717f6de06	Thank understanding highly would revert with...	1	1	1	1	1	1
6	00024115d4cbde0f	Please nonsense Wikipedia edits considered va...	0	0	0	0	0	0
7	000247e83dcc1211	Dear site horrible	1	1	1	1	1	1

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
8	00025358d4737918	fool believe numbers correct number lies Po...	0	0	0	0	0	0
9	00026d1092fe71cc	Double Redirects fixing double redirects bla...	0	0	0	0	0	0

In [15]:

```
toxic = training[training.toxic == 1]
severe_toxic = training[training.severe_toxic == 1]
obscene = training[training.obscene == 1]
threat = training[training.threat == 1]
insult = training[training.insult == 1]
identity_hate = training[training.identity_hate == 1]
```

In [16]:

```
# this will show the difference in datasets, and why I don't intent on combing both valid
ation and training
toxic_validation = validation[training.toxic == 1]
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: UserWarning: Boolean Seri
es key will be reindexed to match DataFrame index.
```

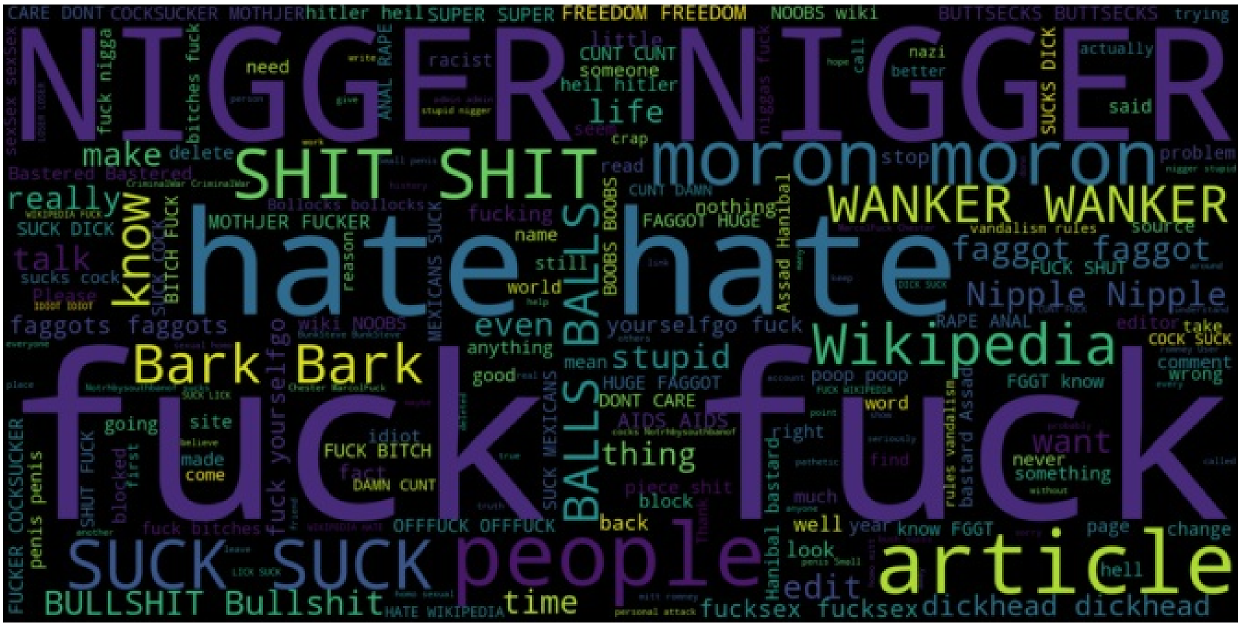
In [17]:

```
import matplotlib.pyplot as plt

def wordcloud_generator(tweet_type):
  neg_string = []
  for t in tweet_type.comment_text:
      neg_string.append(t)
  neg_string = pd.Series(neg_string).str.cat(sep=' ')
  from wordcloud import WordCloud
  wordcloud = WordCloud(width=1600, height=800,max_font_size=300).generate(neg_string)
  plt.figure(figsize=(12,10))
  plt.imshow(wordcloud, interpolation="bilinear")
  plt.axis("off")
  plt.show()
```

In [18]:

```
wordcloud_generator(toxic)
```



In [19]:

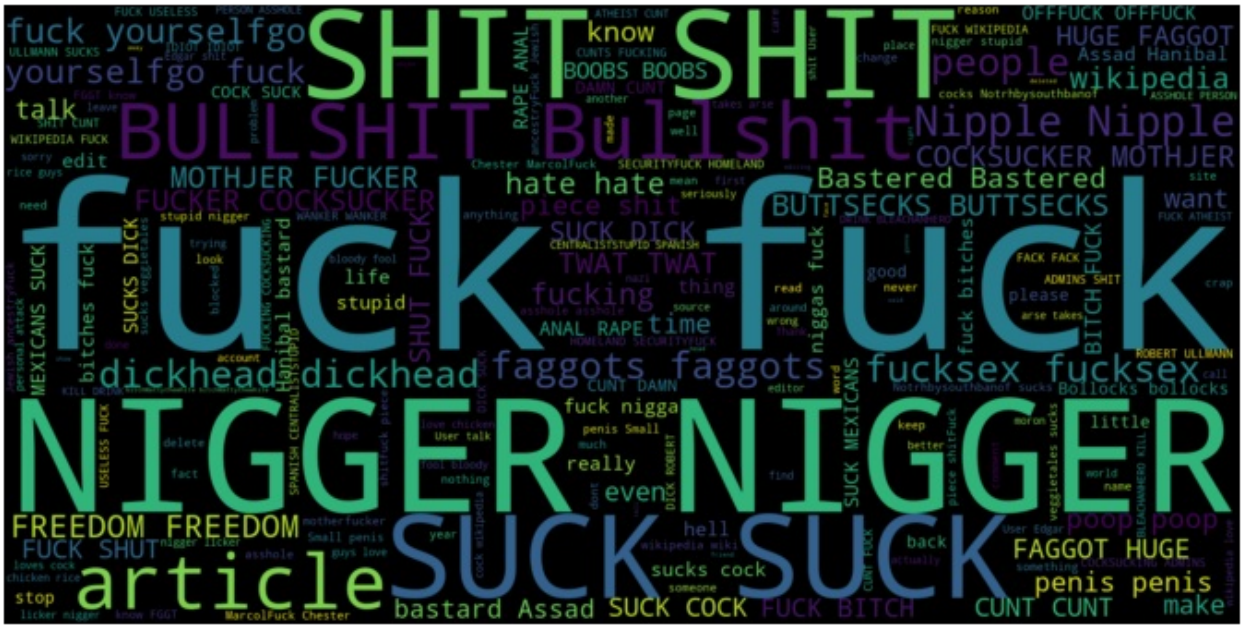
```
#clearly very different than training toxic
wordcloud_generator(toxic_validation)
```

In [20]:

```
wordcloud_generator(severe_toxic)
```

In [21]:

```
wordcloud_generator(obscene)
```



In [22]:

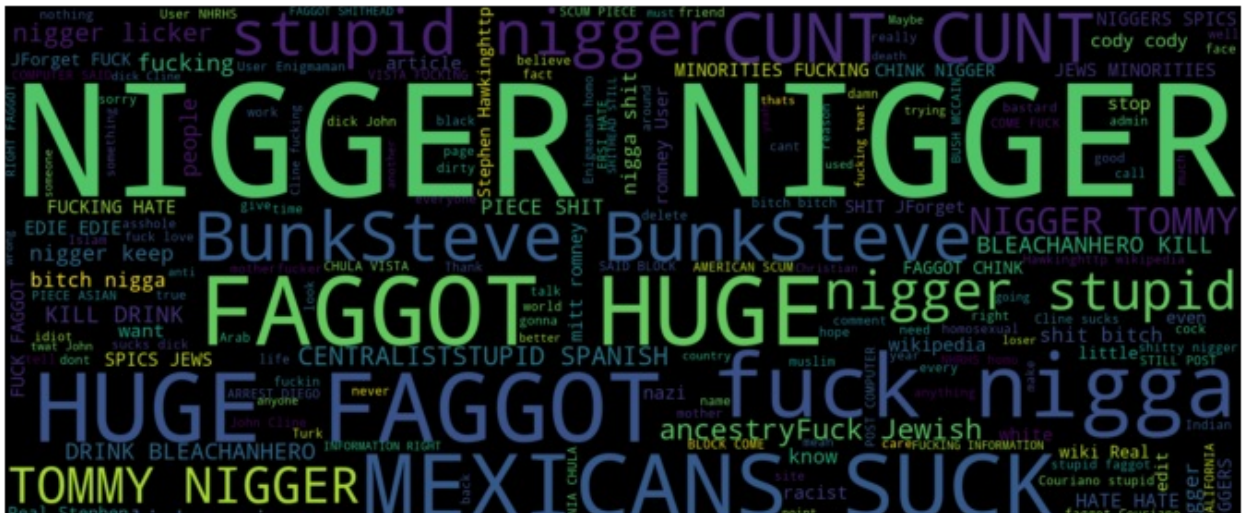
```
wordcloud_generator(threat)
```

In [23]:

```
wordcloud_generator(insult)
```

In [24]:

```
wordcloud_generator(identity_hate)
```



In [25]:

```
train_text = training['comment_text']
print(train_text.shape)
```

```
(159571,)
```

In [26]:

```
import tensorflow_hub as hub
import tensorflow as tf
model = "https://tfhub.dev/google/nnlm-en-dim50/2"
hub_layer = hub.KerasLayer(model, input_shape=[], dtype=tf.string, trainable=True)
hub_layer(train_text[:3])
```

Out[26]:

```
<tf.Tensor: shape=(3, 50), dtype=float32, numpy=
array([[-1.50876045e-01,  9.00597274e-02, -2.92555332e-01,
        -2.92316705e-01, -4.44874726e-02,  1.91590324e-01,
         9.37221795e-02, -1.15236245e-01,  2.54918933e-01,
         9.14832950e-02,  2.61941671e-01,  2.23487750e-01,
         4.10584621e-02, -1.99079648e-01, -1.51285715e-02,
         9.53412205e-02,  2.58471459e-01, -6.62699044e-02,
         1.07833549e-01, -2.28795987e-02, -1.29539505e-01,
         1.07527271e-01, -1.02541551e-01,  1.41762868e-01,
        -3.44812721e-01, -5.04429042e-02, -1.72743827e-01,
        -1.57231808e-01,  2.06909791e-01,  8.52049440e-02,
         4.19160463e-02, -1.16297714e-01, -1.96341679e-01,
        -6.74260408e-02,  6.43082242e-03, -5.59472106e-02,
        -9.76410732e-02,  9.33123939e-03, -3.24791133e-01,
         1.83194965e-01,  6.08906560e-02,  1.59600198e-01,
         1.57664269e-01,  9.28651448e-03, -2.91110039e-01,
         3.04737259e-02, -2.62855679e-01, -1.14653660e-02,
         4.69905496e-01, -1.14612550e-01],
       [ 1.79194845e-02,  8.86705220e-02, -1.18435405e-01,
        -2.50251661e-03, -1.59414917e-01, -3.33138146e-02,
         1.21718697e-01, -1.34936571e-01, -1.75251245e-01,
        -1.21977530e-01,  3.71672153e-01,  1.53097048e-01,
         3.35359611e-02,  1.51222616e-01, -3.98499295e-02,
        -1.96712136e-01, -2.21122801e-01, -2.19928473e-02,
         1.82908863e-01, -3.12797189e-01, -2.36371741e-01,
        -1.77195311e-01,  1.28202096e-01,  4.88120988e-02,
         5.10692932e-02,  9.93956998e-02,  1.70113534e-01,
        -4.98195505e-03,  1.52029395e-01, -1.15033880e-01,
        -1.59197211e-01,  1.71004012e-01, -1.98890548e-02,
         3.10141165e-02,  1.49529830e-01,  3.98692600e-02,
        -5.54314032e-02,  1.26317456e-01, -4.27774061e-03,
        -1.96635783e-01,  1.81764185e-01,  2.74637908e-01,
         1.59178257e-01,  4.67598028e-02, -6.77975193e-02,
        -1.32786706e-01, -2.19241709e-01,  6.78656921e-02,
         3.28906357e-01,  2.24745110e-01],
       [ 1.84072614e-01,  1.24590658e-01,  4.69242930e-02,
        -1.49058290e-02, -3.09239358e-01,  4.59923744e-02,
         9.87137109e-02, -1.64398476e-01, -3.46645385e-01,
         1.16062254e-01,  2.67752677e-01,  6.96456805e-02,
        -1.14691883e-01,  3.48901786e-02, -3.08309793e-01,
         2.18255520e-01, -9.40609276e-02, -2.13600434e-02,
         2.31646091e-01, -2.52506554e-01, -2.99330324e-01,
         2.44463831e-01,  2.80543089e-01, -6.60604239e-02,
        -1.76528424e-01, -1.96415991e-01, -3.44406068e-01,
         4.10415232e-04,  5.28120279e-01, -2.30008930e-01,
         1.94318503e-01,  5.36187477e-02,  8.29566792e-02,
        -3.72426152e-01, -2.32883289e-01, -1.89646482e-01,
         5.17784730e-02,  5.10348827e-02, -1.25595108e-01,
        -6.96755722e-02,  2.33868852e-01,  4.11650062e-01,
         3.59503746e-01,  1.61187872e-01, -2.08950043e-01,
```

```
-2.37224728e-01, -3.24187428e-01, -1.42009705e-01,
 8.54610324e-01,  2.68297613e-01]], dtype=float32)>
```

In [27]:

```
model = tf.keras.Sequential()
model.add(hub_layer)
model.add(tf.keras.layers.Dense(16, activation='relu'))
model.add(tf.keras.layers.Dense(6))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
keras_layer (KerasLayer)     (None, 50)                48190600
_________________________________________________________________
dense (Dense)                (None, 16)                816
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 102
=================================================================
Total params: 48,191,518
Trainable params: 48,191,518
Non-trainable params: 0
_________________________________________________________________
```

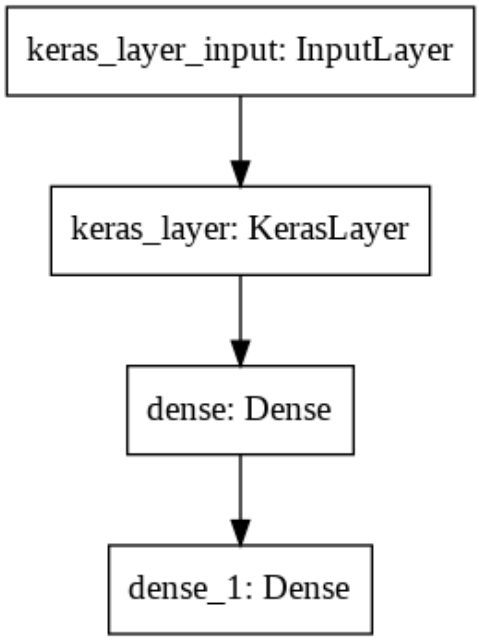
In [28]:

```
model.compile(optimizer='adam',
              loss=tf.losses.BinaryCrossentropy(from_logits=True),
              metrics=[tf.metrics.BinaryAccuracy(threshold=0.0, name='accuracy')])
```

In [29]:

```
from tensorflow.keras.utils import plot_model
plot_model(model, to_file='model.png')
```

Out[29]:



In [30]:

```
history = model.fit(train_text, training_labels, validation_split = 0.15, epochs=10, batch_size=32)
```

```
Epoch 1/10
3990/3990 [==============================] - 119s 29ms/step - loss: 0.1442 - accuracy: 0.9544 - val_loss: 0.0616 - val_accuracy: 0.9795
Epoch 2/10
3990/3990 [==============================] - 116s 29ms/step - loss: 0.0464 - accuracy: 0.9835 - val_loss: 0.0632 - val_accuracy: 0.9800
Epoch 3/10
```

```
Epoch 3/10
3990/3990 [==============================] - 116s 29ms/step - loss: 0.0342 - accuracy: 0.
9873 - val_loss: 0.0709 - val_accuracy: 0.9792
Epoch 4/10
3990/3990 [==============================] - 116s 29ms/step - loss: 0.0253 - accuracy: 0.
9908 - val_loss: 0.0827 - val_accuracy: 0.9785
Epoch 5/10
3990/3990 [==============================] - 116s 29ms/step - loss: 0.0196 - accuracy: 0.
9933 - val_loss: 0.0964 - val_accuracy: 0.9775
Epoch 6/10
3990/3990 [==============================] - 117s 29ms/step - loss: 0.0153 - accuracy: 0.
9948 - val_loss: 0.1104 - val_accuracy: 0.9770
Epoch 7/10
3990/3990 [==============================] - 117s 29ms/step - loss: 0.0124 - accuracy: 0.
9960 - val_loss: 0.1270 - val_accuracy: 0.9763
Epoch 8/10
3990/3990 [==============================] - 117s 29ms/step - loss: 0.0098 - accuracy: 0.
9970 - val_loss: 0.1378 - val_accuracy: 0.9761
Epoch 9/10
3990/3990 [==============================] - 117s 29ms/step - loss: 0.0088 - accuracy: 0.
9973 - val_loss: 0.1455 - val_accuracy: 0.9752
Epoch 10/10
3990/3990 [==============================] - 116s 29ms/step - loss: 0.0074 - accuracy: 0.
9978 - val_loss: 0.1560 - val_accuracy: 0.9749
```

In [31]:

```
#mount to google drive so we can save our model there
from google.colab import drive
drive.mount('/content/drive')
path = path = F"/content/drive/My Drive/toxiccommentmodel2"
```

```
Mounted at /content/drive
```

In [35]:

```
model.save(path)
```

```
INFO:tensorflow:Assets written to: /content/drive/My Drive/toxiccommentmodel2/assets
```

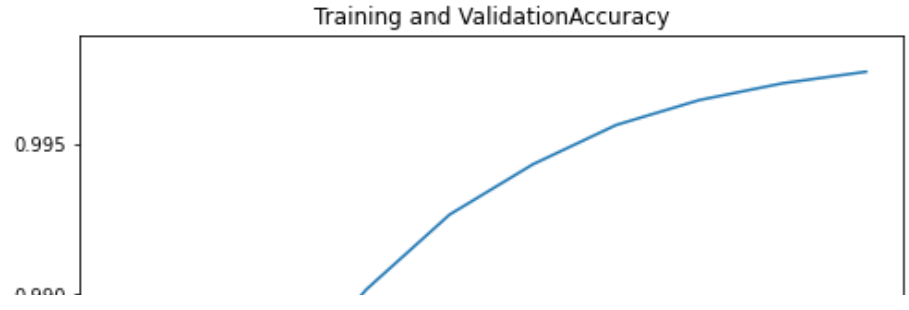
INFO:tensorflow:Assets written to: /content/drive/My Drive/toxiccommentmodel2/assets

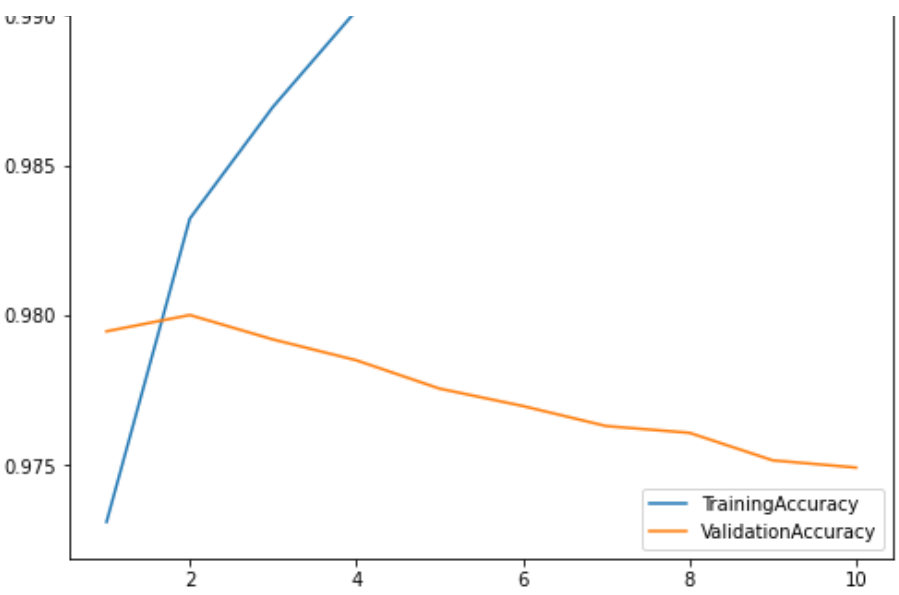
In [41]:

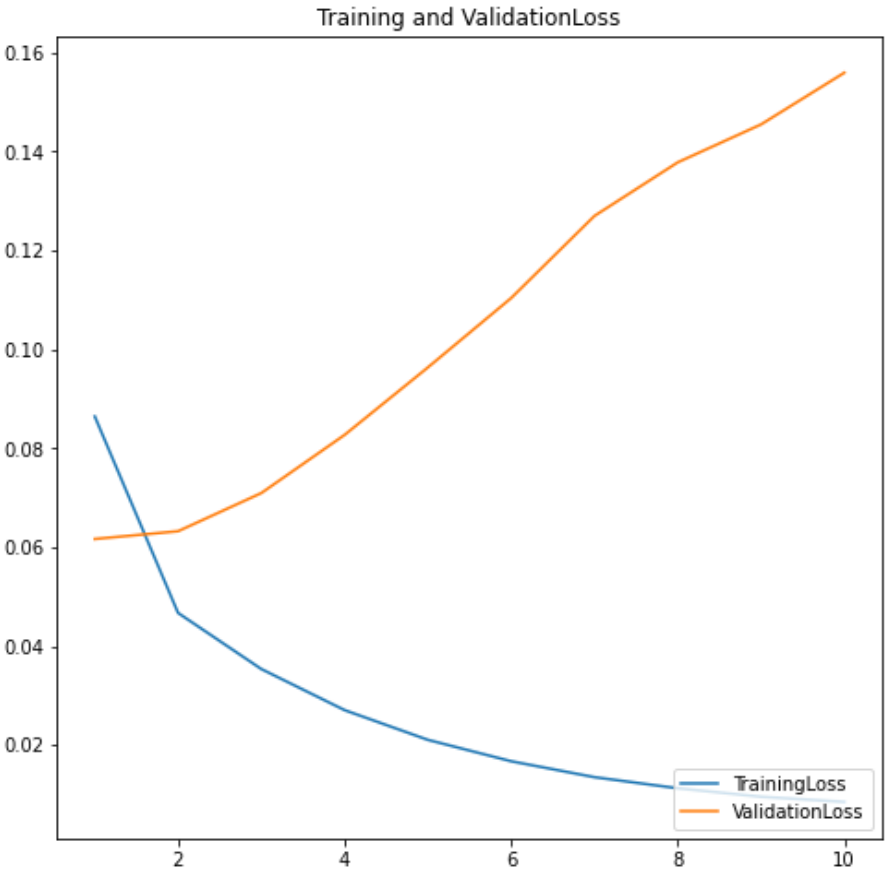
```
import matplotlib.pyplot as plt
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(1, 11)
def plot(x, y, z):
  plt.figure(figsize=(8, 8))
  #plt.subplot(1, 2, 1)
  plt.plot(epochs_range, x, label=('Training' + z))
  plt.plot(epochs_range, y, label= ('Validation' + z))
  plt.legend(loc='lower right')
  plt.title('Training and Validation' + z)
  plt.show()
plot(acc, val_acc, 'Accuracy')
plot(loss, val_loss, 'Loss')
```







In [67]:

```
def feedback(input):
  print("\nresults for","'",input,"'")
  predictions = model.predict(np.expand_dims(input, 0))
  predictions = tf.constant(predictions)
  predictions = (tf.keras.activations.sigmoid(predictions)).numpy()
  predictions = predictions * 100
  for i in range(len(predictions[0])):
    print(" ", classes[i], predictions[0][i])

  print('The model identifies the text as', end = " ")

  for x in range (0,len(classes)):
    y = []
    if ((predictions[0])[x]) > 50:
      print((classes[x]).upper(), end = " ")
    if ((predictions[0])[x]) < 50:
      y.append(x)
      y= np.hstack(y)
      if not y:
        print("most likely not containing hate content")
```

```
# EXTREMELEY VULGAR TEXT BELOW!
# I intentionally chose text that does not use any explicit words that would be marked by
a key-word bot

examples = [
            'shut up curry monkey',
            'jews are a virus',
            'the file wont download'
            ]

for i in range(len(examples)):
  feedback(examples[i])
  print("\n")
```

```
results for ' shut up curry monkey '
  toxic 99.98999
  severe_toxic 0.0033222495
  obscene 1.7957094
  threat 7.6399534e-05
  insult 97.31152
  identity_hate 0.17763154
The model identifies the text as TOXIC INSULT


results for ' jews are a virus '
  toxic 81.32488
  severe_toxic 0.31375593
  obscene 2.9781437
  threat 0.13528347
  insult 8.382479
  identity_hate 90.67519
The model identifies the text as TOXIC IDENTITY_HATE


results for ' the file wont download '
  toxic 4.8895545e-09
  severe_toxic 3.1642234e-08
  obscene 3.7257358e-08
  threat 1.5080008e-07
  insult 4.637114e-09
  identity_hate 1.4796031e-08
The model identifies the text as most likely not containing hate content
```