In [2]:

```
import numpy as np
```

In [5]:

```
UnScaledX = np.loadtxt ('UnscaledX.txt')
ScaledX = np.loadtxt ('ScaledX.txt')
```

In [3]:

```
Y = [l.strip() for l in open ('Y.txt').xreadlines()]
```

In [9]:

```
print ScaledX.shape
print UnScaledX.shape
len(Y)
```

```
(3698, 20700)
(3698, 20700)
```

Out[9]:

```
3698
```

In [10]:

```
Z = list(zip(ScaledX, Y))
```

In [14]:

```
from random import shuffle
shuffle(Z)
X, Y = zip(*Z)
```

In [15]:

```
TrCutOff = int(len(Y)*0.8-1)
TeCutoff = TrCutOff - len(Y)
print TrCutOff, TeCutoff
```

```
2957 -741
```

In [101]:

```
X_train = np.array(X[:TrCutOff])
Y_train =  np.array(Y[:TrCutOff])
X_test =  np.array(X[TeCutoff:])
Y_test =  np.array(Y[TeCutoff:])

print X_train.shape
print Y_train.shape
print X_test.shape
print Y_test.shape
```

```
(2957, 20700)
(2957,)
(741, 20700)
(741,)
```

In [102]:

```
MaxFeatsWithPadd = 69
WordVecDims = 300
```

In [103]:

```
from collections import Counter
from keras.utils import np_utils
X_train = X_train.reshape(X_train.shape[0],1, MaxFeatsWithPadd, WordVecDims)
X_test = X_test.reshape(X_test.shape[0],1, MaxFeatsWithPadd, WordVecDims)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
print('X_train shape:', X_train.shape)
print('X_test shape:', X_test.shape)
print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')
```

```
('X_train shape:', (2957, 1, 69, 300))
('X_test shape:', (741, 1, 69, 300))
(2957, 'train samples')
(741, 'test samples')
```

In [104]:

```
def MakeYAsNum (Y):
    Ytmp = []
    for y in Y:
        if 'positive' == y:
            Ytmp.append (1)
        elif 'negative' == y:
            Ytmp.append(2)
        elif 'conflict' == y:
            Ytmp.append(3)
        else:
            Ytmp.append(0)
    return Ytmp
```

In [105]:

```
Y_train = MakeYAsNum(Y_train)
Y_test = MakeYAsNum(Y_test)
```

In [106]:

```
print(Counter(Y_train))
print(Counter(Y_test))
```

```
Counter({1: 1735, 2: 639, 0: 522, 3: 61})
Counter({1: 428, 2: 168, 0: 115, 3: 30})
```

In [107]:

```
Y_train = np_utils.to_categorical(Y_train, nb_classes)
Y_test = np_utils.to_categorical(Y_test, nb_classes)
```

In [108]:

```
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D, MaxPooling
from keras.utils import np_utils
import keras
```

In [123]:

```
nb_filters = 100
nb_pool = 3
model = Sequential()

model.add(Convolution2D(nb_filters, 3, 3,
                        border_mode='valid',
                        input_shape=(1, 69, 300)))
model.add(Activation('relu'))
#model.add(Convolution2D(nb_filters, 3, 3))
#model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy',
              optimizer='adadelta',
              metrics=['accuracy'])
```

In [124]:

```
model.fit(X_train, Y_train, batch_size=10, nb_epoch=5,
          verbose=1, validation_data=(X_test, Y_test))
T0 = time.time()
score = model.evaluate(X_test, Y_test, verbose=1)
print (time.time() - T0)
```

```
INFO (theano.gof.compilelock): Refreshing lock /home/annamalai/.t
heano/compiledir_Linux-3.13--generic-x86_64-with-Ubuntu-12.04-pre
cise-x86_64-2.7.3-64/lock_dir/lock

Train on 2957 samples, validate on 741 samples
Epoch 1/5
2957/2957 [==============================] - 767s - loss: 1.0528
- acc: 0.5742 - val_loss: 1.0148 - val_acc: 0.5870
Epoch 2/5
2957/2957 [==============================] - 774s - loss: 0.9402
- acc: 0.6101 - val_loss: 0.9782 - val_acc: 0.6046
Epoch 3/5
2957/2957 [==============================] - 798s - loss: 0.8459
- acc: 0.6581 - val_loss: 0.8702 - val_acc: 0.6437
Epoch 4/5
2957/2957 [==============================] - 799s - loss: 0.7330
- acc: 0.7109 - val_loss: 0.8445 - val_acc: 0.6748
Epoch 5/5
2957/2957 [==============================] - 1180s - loss: 0.6607
- acc: 0.7504 - val_loss: 0.8142 - val_acc: 0.6883

-----------------------------------------------------------------
----------
NameError                                 Traceback (most recent
call last)
<ipython-input-124-4295289060f0> in <module>()
      1 model.fit(X_train, Y_train, batch_size=10, nb_epoch=5,
      2           verbose=1, validation_data=(X_test, Y_test))
----> 3 T0 = time.time()
      4 score = model.evaluate(X_test, Y_test, verbose=1)
      5 print (time.time() - T0)

NameError: name 'time' is not defined
```

In [59]:

```
print model.layers
```

```
[<keras.layers.convolutional.Convolution2D object at 0xacad250>,
<keras.layers.core.Activation object at 0xacad2d0>, <keras.layer
s.core.Dropout object at 0x651c73d0>, <keras.layers.core.Flatten
object at 0x6071c9d0>, <keras.layers.core.Dense object at 0x5ec22
a90>, <keras.layers.core.Activation object at 0x58387290>, <kera
s.layers.core.Dropout object at 0x58387950>]
```