(https://www.nvidia.com/dli)

비디오 자막 생성(Video Captioning)

실습을 시작하기 위해 비디오 자막 생성에 대해 알아보겠습니다. 우리는 이미지 자막 생성 위해 사용한 것과 유사한 기술을 적용할 것입니다. RNN에 대한 입력을 생성하기 위해 feature vector 및 자막을 병합합니다. 비디오의 경우, 각 비디오 클립의 프레임에서 `fc7` feature vector의 *평균*을 사용합니다. Microsoft Research Video Description Corpus(MSVD) (https://www.microsoft.com/en-us/download/details.aspx (https://www.microsoft.com/en-us/download/details.aspx)? ID=52422)[2] 데이터셋에는 각각 10개 이상의 자막이 있는 약 2000개의 비디오 클립이 포함되어 있습니다.

In [14]:

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import inspect
import time

import numpy as np
import tensorflow as tf
from tensorflow.python.framework import ops
from tensorflow.python.framework import dtypes
#import reader
import collections
import os
import re
import json

import matplotlib.pyplot as plt

from scipy import ndimage
from scipy import misc
import sys
sys.path.insert(0, '/dli/data/mdt/models/slim')

slim=tf.contrib.slim
from nets import vgg

from preprocessing import vgg_preprocessing

%matplotlib inline
!nvidia-smi
```

```
Mon Feb 24 13:59:15 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 396.26                 Driver Version: 396.26                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:1E.0 Off |                    0 |
| N/A   60C    P0    55W / 149W |  10975MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```

MSVD 데이터

이제 데이터를 살펴보겠습니다. 첫째, 우리의 자막과 비디오 클립입니다.** "(CaptionsandMovies[])"**로 인덱스를 다른 번호로 자유롭게 변경해 보십시오.

In [15]:

```
caption_file = open('MSVDCaptions_lower_nochars.txt', 'r')
data=caption_file.readlines()
caption_file.close()
CaptionsandMovies=[row.split(', ') for row in data]
print(CaptionsandMovies[0])
```

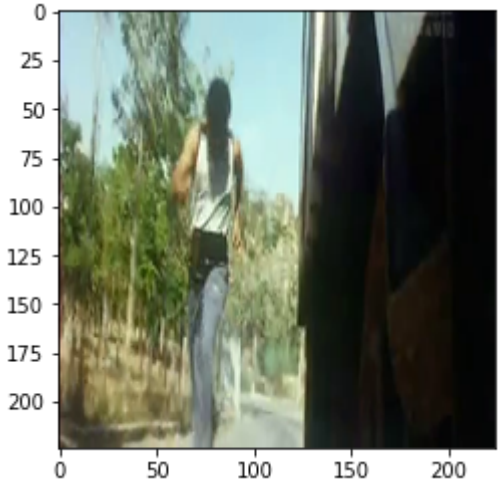
```
['mv89psg6zh4_33_46', 'a bird in a sink keeps getting under the running water from a
faucet ₩n']
```

In [16]:

```
clipnum=0

import os
from glob import glob
ListofClips = glob('/dli/data/mdt/msvd/frames_224/*')
print(ListofClips[clipnum])
clip_frames = [f for f  in os.listdir(ListofClips[0])]
print(len(clip_frames),' frames from this clip are used to train the network')
print('One Frame from clip ',os.path.basename(ListofClips[0]),' is displayed below')
one_image=ndimage.imread(ListofClips[clipnum]+'/'+clip_frames[0])
#resize for vgg network
resize_img=misc.imresize(one_image,[224,224])

#Show image
plt.imshow(resize_img)

#Display Captions for Video clips
MovieCaption=[vid_id[1] for vid_id in CaptionsandMovies if vid_id[0]==ListofClips[clipnum][22:]]
print('captions for this clip')
print(MovieCaption)
clipnum=0

import os
from glob import glob
ListofClips = glob('/dli/data/mdt/msvd/frames_224/*')
print(ListofClips[clipnum])
clip_frames = [f for f  in os.listdir(ListofClips[0])]
print(len(clip_frames),' frames from this clip are used to train the network')
print('One Frame from clip ',os.path.basename(ListofClips[0]),' is displayed below')
one_image=ndimage.imread(ListofClips[clipnum]+'/'+clip_frames[0])
#resize for vgg network
resize_img=misc.imresize(one_image,[224,224])

#Show image
plt.imshow(resize_img)

#Display Captions for Video clips
MovieCaption=[vid_id[1] for vid_id in CaptionsandMovies if vid_id[0]==os.path.basename(ListofClips
print('captions for this clip')
print(MovieCaption)
```

```
/dli/data/mdt/msvd/frames_224/2YhDTpzxd3c_223_232
1  frames from this clip are used to train the network
One Frame from clip  2YhDTpzxd3c_223_232  is displayed below
captions for this clip
[]
/dli/data/mdt/msvd/frames_224/2YhDTpzxd3c_223_232
1  frames from this clip are used to train the network
One Frame from clip  2YhDTpzxd3c_223_232  is displayed below
captions for this clip
['a man outruns a truck driving down the street ₩n', 'a man runs ahead of a truck
₩n', 'a man is running alongside a truck and passes it ₩n', 'a man is running down t
he road while a truck drives along beside him ₩n', 'a man is running with a bus ₩n',
'a man is chasing a truck ₩n', 'a man is running next to a truck that is driving dow
n a dirt road ₩n', 'a man is overtaking a truck by running ₩n', 'a man outruns a tru
ck ₩n', 'a man is running past a truck ₩n', 'a man is running to get infront of a mo
```

```
ving vehicle ₩n', 'a man is running after a truck ₩n', 'a man running on the road be
side a speeding truck outruns the vehicle and stands in front of it ₩n', 'the man ra
ced against the truck and got in front of it ₩n', 'the man is racing the truck ₩n',
'a man is running after a truck and runs in front of it ₩n', 'a man runs alongside a
truck ₩n', 'the man beat the truck by running next to it ₩n', 'a man is running next
to a truck ₩n', 'in the tamil movie vijay and poomiga talking and vijay practice exe
rcise₩n', 'the person is running a way₩n', 'the traning for do u know who is pavan k
alyan in andhra he is andhra biggest image person₩n', 'a man is running alongside a
truck ₩n', 'a man on foot races a truck and wins ₩n', 'a man running fast to chase a
truck ₩n', 'a man running hard behind a truck ₩n', 'the man raced against a truck an
d got in front of the truck ₩n', 'the person is running ₩n', 'vijay acting is too go
od₩n', 'a person is running₩n', 'a scene from an indian film ₩n']
```



We'll create a mean vector of a single clip by running each frame through VGG, a pretrained image classification model. This will generate a high-level representation of each frame from layer fc7.

미리 구성된 이미지 분류 모델인 VGG를 통해 각 프레임을 실행함으로써 단일 클립의 평균 벡터를 만들 것입니다. 이렇게 하면 `fc7` 계층에서 각 프레임의 수준 높은 표현(representation)이 생성됩니다

In [17]:

```
#Create a mean vector of a single video clip

TRAIN_DATA_PATH='/dli/data/mdt/msvd/'
## Read Training files
CLIPOFINTEREST='KpmVL4ANieA_0_9'
one_clip_of_interest = [f for f  in os.listdir(TRAIN_DATA_PATH+'frames_224/'+CLIPOFINTEREST)]
clip_feature_vectors={}
for frame in one_clip_of_interest:
    tf.reset_default_graph()
    one_image=ndimage.imread('/dli/data/mdt/msvd/frames_224/'+CLIPOFINTEREST+'/'+frame)
    #resize for vgg network
    resize_img=misc.imresize(one_image,[224,224])
    if len(one_image.shape)!= 3: #Check to see if the image is grayscale if True mirror colorband
        resize_img=np.asarray(np.dstack((resize_img, resize_img, resize_img)), dtype=np.uint8)
    #image_size = vgg.vgg_16.default_image_size
    processed_image = vgg_preprocessing.preprocess_image(resize_img, 224, 224, is_training=False)
    processed_images  = tf.expand_dims(processed_image, 0)
    network,endpts= vgg.vgg_16(processed_images, is_training=False)

    init_fn = slim.assign_from_checkpoint_fn(os.path.join('/dli/data/mdt/mscoco/vgg_16.ckpt'),slim.g
    sess = tf.Session()
    init_fn(sess)
    NETWORK,ENDPTS=sess.run([network,endpts])
    sess.close()
    clip_feature_vectors[frame]=ENDPTS['vgg_16/fc7'][0][0][0]
```

```
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
INFO:tensorflow:Restoring parameters from /dli/data/mdt/mscoco/vgg_16.ckpt
```

In [18]:

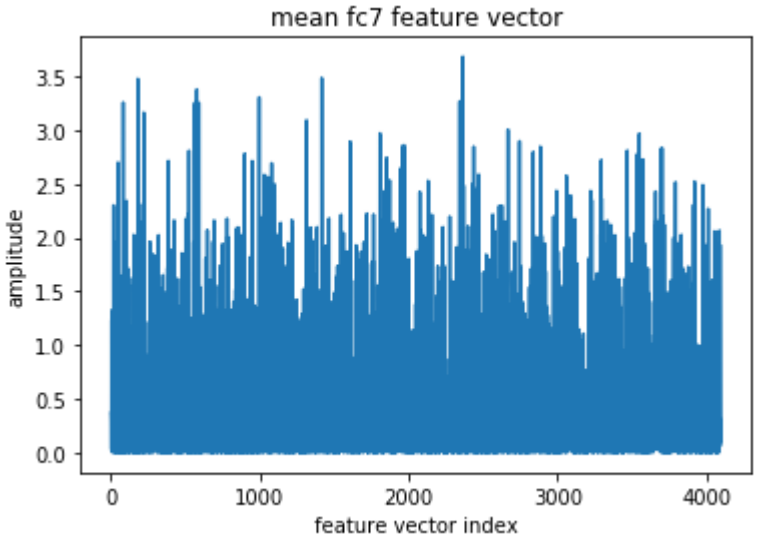
```
numframes=len(clip_feature_vectors.keys())
meanfc7vector=np.zeros([numframes,4096])
for i in range(numframes):
    i=i+1
    meanfc7vector[i-1,:]=clip_feature_vectors['frame_'+'%06d' %i+'_224.bmp'].reshape(1,4096)

meanfc7vector=np.mean(meanfc7vector,axis=0)
```

In [19]:

```
plt.plot(meanfc7vector)
plt.xlabel('feature vector index')
plt.ylabel('amplitude')
plt.title('mean fc7 feature vector')
```

Out[19]:

```
<matplotlib.text.Text at 0x7fce34cd8b10>
```



이는 이미지 자막 노트북에서 RNN의 context로 사용했던 feature vector처럼 보일 것입니다. 우리의 RNN에게도 똑같이 보일 것이기 때문에, 이 시점에서 우리가 사용할 워크플로는 이미지에서 사용한 워크플로우를 동일하게 사용합니다.

시작하기 전에 잠시 시간을 내어 GPU 메모리를 비우고, 교육에 필요한 것만 다시 로드합니다. 이것은 실습중, 메모리 부족문제가 발생할 위험을 제거해 줍니다. 메모리 관리 시간을 줄이고, 실습에 더 집중하기 원한다면 더 높은 성능의 GPU를 고려해 보시기 바랍니다.:)

계속하기 전에 이 노트북 상단에 있는 메뉴에서 커널 -> 재시작(Kernel -> Restart)을 선택하십시오.

이렇게 하면 교육 중 사용할 일부 작업이 지워지게 됩니다. 다음 셀을 실행하여 라이브러리와 데이터를 다시 불러내시기 바랍니다.

학습 설정하기 (Training Setup)

In [20]:

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import inspect
import time

import numpy as np
import tensorflow as tf
from tensorflow.python.framework import ops
from tensorflow.python.framework import dtypes
#import reader
import collections
import os
import re
import json

import matplotlib.pyplot as plt

from scipy import ndimage
from scipy import misc
import sys
sys.path.insert(0, '/dli/data/mdt/models/slim')

slim=tf.contrib.slim
from nets import vgg

from preprocessing import vgg_preprocessing

%matplotlib inline

caption_file = open('MSVDCaptions_lower_nochars.txt', 'r')
data=caption_file.readlines()
caption_file.close()
CaptionsandMovies=[row.split(', ') for row in data]
```

이미지 평균에서 high-level feature vector 로드하기

이 단계는 여러분을 위해 이미 완료되어 있습니다. 벡터가 저장된 위치를 기록해 두십시오.

In [21]:

```
#A npy file of all of the mean feature vectors has been created
file_loader=np.load('/dli/data/mdt/msvd/train_msvd_mean_fc7.npy').tolist()
print('Number of videos in the training set: ', len(file_loader.keys()))
```

```
Number of videos in the training set:  1773
```

이제 feature map들을 해당 자막에 맞춰 정렬할 것입니다. 평균 feature vector로 구성된 npy 파일이 작성되었습니다. 이제 이러한 feature map과 각각의 자막을 병합합니다.

자막들과 비디오들을 정렬시키기(Aligning captions with videos)

In [22]:

```
#Create 3 lists image_id, feature maps, and captions.
video_id_key=[]
feature_maps_to_id=[]
caption_to_id=[]
for observed_vid in file_loader.keys():
    for k in range(len(CaptionsandMovies)):
        if CaptionsandMovies[k][0]==observed_vid:
            video_id_key.append([observed_vid])
            feature_maps_to_id.append(file_loader[observed_vid])
            caption_to_id.append(CaptionsandMovies[k][1])


print('number of captions and datapoints',len(caption_to_id))
```

```
number of captions and datapoints 63648
```

자막 인코딩 하기 및 딕셔너리(dictionary)에 제한 설정하기

원-핫 인코딩(one-hot encoding)을 사용하여 언어 기반 데이터를 토큰화 할 때 딕셔너리의 길이는 각 워드 벡터 길이와 동일합니다. 딕셔너리를 네개까지로 제한하면, 대부분의 경우 모델 성능에 크게 영향을 주지 않으면서 훈련 속도가 크게 빨라집니다. 이것은 흔한 방법 입니다.

In [23]:

```
#Now we will encode our captions and set a dictionary limit

num_steps=20
#######################################################################
##Create a list of all of the sentences.
DatasetWordList=[]
for dataset_caption in caption_to_id:
        DatasetWordList+=dataset_caption.split()

#Determine number of distinct words
distinctwords=collections.Counter(DatasetWordList)
#Order words
count_pairs = sorted(distinctwords.items(), key=lambda x: (-x[1], x[0])) #ascending order
words, occurence = list(zip(*count_pairs))
DictionaryLength=occurence.index(4) #index for words that occur 4 times or less
words=['PAD','UNK','EOS']+list(words[:DictionaryLength])
word_to_id=dict(zip(words, range(len(words))))
#####################  Tokenize Sentence #######################
Tokenized=[]
for full_words in caption_to_id:
        EmbeddedSentence=[word_to_id[word] for word in full_words.split() if word in word_to_id]+[
        #Pad sentences that are shorter than the number of steps
        if len(EmbeddedSentence)<num_steps:
            b=[word_to_id['PAD']]*num_steps
            b[:len(EmbeddedSentence)]=EmbeddedSentence
        if len(EmbeddedSentence)>num_steps:
            b=EmbeddedSentence[:num_steps]
        if len(b)==EmbeddedSentence:
            b=EmeddedSentence
        b=[word_to_id['UNK'] if x>=DictionaryLength else x for x in b] #turn all words used 4 time
        #print(b)
        Tokenized+=[b]

print("Number of words in this dictionary ", len(words))
```

```
Number of words in this dictionary  3745
```

이미지 자막에 사용한 것과 동일한 모델을 사용할 수 있습니다.

문제 해결 지원

종종 실습 중에 다음과 같은 오류 메시지가 뜨기도 합니다.

"InternalError: Dst tensor is not initialized"

"ResourceExhaustedError: OOM when allocating tensor with shape"

이 메시지가 표시되면 Python 커널을 다시 시작해야 할 수 있습니다. 이 커널은 이 Juptyer 노트북의 상단에 있는 메뉴 모음으로 이동하여 커널을 선택한 다음 다시 시작 (Restart)을 선택하여 수행할 수 있습니다. Python 커널을 다시 시작할 때 모든 변수를 다시 정의해야 합니다. 이러한 문제가 발생하면 위의 "학습 설정 (Training Setup)"으로 돌아가서 다시 시도하십시오.

In [24]:

```
## Remember that we use the previous words and the feature vector as inputs to predict future words.

def data_queue(TrainingInputs,FeatureVectors):
    train_input_queue = tf.train.slice_input_producer(
                    [TrainingInputs, np.asarray(FeatureVectors)],num_epochs=10000,
                                shuffle=True) #False before

    ##Set our train data and label input shape for the queue

    TrainingInputs=train_input_queue[0]
    FeatureVectors=train_input_queue[1]
    TrainingInputs.set_shape([num_steps])
    FeatureVectors.set_shape([len(feature_maps_to_id[0])]) #fc7 is 4096
    ##LabelInput.set_shape([num_steps])
    min_after_dequeue=1000000
    capacity = min_after_dequeue + 3 * batch_size
    #input_x, target_y
    tokenized_caption, input_feature_map = tf.train.batch([TrainingInputs, FeatureVectors],
                                            batch_size=batch_size,
                                            capacity=capacity,
                                            num_threads=6)
    return tokenized_caption, input_feature_map

def rnn_model(Xconcat,input_keep_prob,output_keep_prob,num_layers,num_hidden):
#Create a multilayer RNN
#reuse=False for training but reuse=True for sharing
    layer_cell=[]
    for _ in range(num_layers):
        lstm_cell = tf.contrib.rnn.LSTMCell(num_units=num_hidden, state_is_tuple=True)
        lstm_cell = tf.contrib.rnn.DropoutWrapper(lstm_cell,
                                        input_keep_prob=input_keep_prob,
                                        output_keep_prob=output_keep_prob)
        layer_cell.append(lstm_cell)

    cell = tf.contrib.rnn.MultiRNNCell(layer_cell, state_is_tuple=True)
    outputs, last_states = tf.contrib.rnn.static_rnn(
        cell=cell,
        dtype=tf.float32,
        inputs=tf.unstack(Xconcat))

    output_reshape=tf.reshape(outputs, [batch_size*(num_steps),num_hidden]) #[12==batch_size*num_st
    pred=tf.matmul(output_reshape, variables_dict["weights_mscoco"]) +variables_dict["biases_mscoco"
    return pred
```

미리 학습된 RNN을 로드하고 조금씩 다시 학습해 보십시오. 미리 학습 완료된 RNN은 여기 : /dli/data/mdt/msvd/video_4096_6model_iter99999 에 저장되어 있습니다. 또한 모델 생성을 위해서, 이미지 자막(Image Captioning) 실습에서 사용했던 함수를 사용하겠습니다.

```
    saver.restore(sess, PRETRAINED_MODEL)
```

Loss가 어떻게 됩니까?

In [25]:

```
tf.reset_default_graph()
##########################################################################################################
# Parameters
num_hidden=4096
num_steps=num_steps
dict_length=len(words)
batch_size=4
num_layers=2
loss_msvd=[]
train_lr=0.00001
##########################################################################################################
TrainingInputs=Tokenized
FeatureVectors=feature_maps_to_id

tokenized_caption, input_feature_map=data_queue(TrainingInputs,FeatureVectors)
## Make Variables

lr = tf.placeholder(tf.float32, shape=[])
#tf.get_variable_scope().reuse_variables()

variables_dict = {
    "weights_mscoco":tf.Variable(tf.truncated_normal([num_hidden,dict_length],
                                                     stddev=1.0,dtype=tf.float32),name="weights_msc
    "biases_mscoco": tf.Variable(tf.truncated_normal([dict_length],
                                                     stddev=1.0,dtype=tf.float32), name="biases_msc




TrainInput=tf.constant(word_to_id['PAD'],shape=[batch_size,1],dtype=tf.int32)
#Pad the beginning of our caption. The first step now only has the image feature vector. Drop the la
#to timesteps to 20
TrainInput=tf.concat([tf.constant(word_to_id['PAD'],shape=[batch_size,1],dtype=tf.int32),
                      tokenized_caption],1)[:,:-1]
X_one_hot=tf.nn.embedding_lookup(np.identity(dict_length), TrainInput) #[batch,num_steps,dictionary_
#ImageFeatureTensor=input_feature_map
Xconcat=tf.concat([input_feature_map+tf.zeros([num_steps,batch_size,4096]),
                                                tf.unstack(tf.to_float(X_one_hot),num_steps,1)],2)#


#the full caption is now the target sentence
y_one_hot=tf.unstack(tf.nn.embedding_lookup(np.identity(dict_length), tokenized_caption),num_steps,1

y_target_reshape=tf.reshape(y_one_hot,[batch_size*num_steps,dict_length])
pred=rnn_model(Xconcat,1.0,1.0,num_layers,num_hidden)

cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=pred, labels=y_target_reshape)
optimizer = tf.train.AdamOptimizer(learning_rate=lr,epsilon=1.0)


gvs = optimizer.compute_gradients(cost,  aggregation_method = tf.AggregationMethod.EXPERIMENTAL_TREE
capped_gvs = [(tf.clip_by_value(grad, -10., 10.), var) for grad, var in gvs]
train_op=optimizer.apply_gradients(capped_gvs)

saver = tf.train.Saver()
init_op = tf.group(tf.global_variables_initializer(),
                   tf.local_variables_initializer())
```

```
with tf.Session() as sess:

    sess.run(init_op)
    coord = tf.train.Coordinator()
    threads = tf.train.start_queue_runners(coord=coord)
    saver.restore(sess, '/dli/data/mdt/msvd/video_4096_6model_iter99999')
    for i in range(100):

        loss,y_pred,target_caption,_,ftmap=sess.run([cost,pred,tokenized_caption,train_op,input_feat
                                                     feed_dict={lr:train_lr})

        #print loss
        if i% 10==0:
            print("iteration: ",i, "loss: ",loss)


    MODEL_NAME='/dli/data/mdt/msvd/video_4096model_iter'+str(i)
    saver.save(sess, MODEL_NAME)
    print('Saved network ',MODEL_NAME)
    print("Done Training")
    coord.request_stop()
    coord.join(threads)
    sess.close()
```

```
INFO:tensorflow:Restoring parameters from /dli/data/mdt/msvd/video_4096_6model_iter9
9999
iteration:  0 loss:  0.7828371
iteration:  10 loss:  0.7861327
iteration:  20 loss:  0.69346297
iteration:  30 loss:  1.1232129
iteration:  40 loss:  0.8137539
iteration:  50 loss:  0.94670236
iteration:  60 loss:  0.6635798
iteration:  70 loss:  1.074105
iteration:  80 loss:  0.6378349
iteration:  90 loss:  1.0371939
Saved network  /dli/data/mdt/msvd/video_4096model_iter99
Done Training
```

신경망이 자막의 다음 단어를 얼마나 잘 예측할 수 있는지 아래 함수를 사용하여 알 수 있습니다. 배치의 인덱스(0과 3)을 사용하여 마지막 인덱스에서 단일 영상과 해당 자막을 확인하여 평가할 수 있습니다.

저장된 시점의 신경망 상태에 따라 불완전하고 일관성이 없으며 때로는 부적절한 자막이 생성될 수 있다는 것을 기억하십시오.

In [26]:

```
def eval_prediction(batch_id,batch_size,words,target_catpion,predicted_caption, ftmap):
    TARGETSENTENCE=[words[ind] for ind in target_caption[batch_id]]
    PREDICTEDSENTENCE=[words[ind] for ind in np.argmax(y_pred[batch_id::batch_size],1)]
    VIDEOCLIPNAME=[x for x in file_loader.keys() if np.array_equal(ftmap[batch_id],file_loader[x])]
    return TARGETSENTENCE,PREDICTEDSENTENCE, VIDEOCLIPNAME[0]

batch_id=100
t,p,v=eval_prediction(batch_id,batch_size,words,target_caption,y_pred,ftmap)
#print('Ground Truth Words')
#print(t)
print('Predicted Words')
print(p)
exp_image=ndimage.imread('/dli/data/mdt/msvd/frames_224/'+v+'/frame_000003_224.bmp')
plt.imshow(exp_image)
plt.title('video clip name:'+v)
```

```
IndexErrorTraceback (most recent call last)
<ipython-input-26-1e26e6c04d93> in <module>()
      6
      7 batch_id=100
----> 8 t,p,v=eval_prediction(batch_id,batch_size,words,target_caption,y_pred,ftmap)
      9 #print('Ground Truth Words')
     10 #print(t)

<ipython-input-26-1e26e6c04d93> in eval_prediction(batch_id, batch_size, words, targ
et_catpion, predicted_caption, ftmap)
      1 def eval_prediction(batch_id,batch_size,words,target_catpion,predicted_capt
ion, ftmap):
----> 2     TARGETSENTENCE=[words[ind] for ind in target_caption[batch_id]]
      3     PREDICTEDSENTENCE=[words[ind] for ind in np.argmax(y_pred[batch_id::bat
ch_size],1)]
      4     VIDEOCLIPNAME=[x for x in file_loader.keys() if np.array_equal(ftmap[ba
tch_id],file_loader[x])]
      5     return TARGETSENTENCE,PREDICTEDSENTENCE, VIDEOCLIPNAME[0]

IndexError: index 100 is out of bounds for axis 0 with size 4
```

저장된 네트워크를 로드한 후 이미지 캡션 실습에서와 같이 검증 이미지에서 캡션을 생성합니다.

방금 만든 미리 학습된 네트워크를 로드하고 검증 클립에 대한 캡션을 생성하는 데 사용합니다. 비디오 클립은 VALDATA 변수를 사용하여 테스트할 수 있습니다. ##FIXME## 을 0에서 196 사이의 값으로 바꿔 사용해 보십시오. 예를 들어 이 변수를 0, VALDATA=0으로 설정할 수 있습니다.

이전 셀에 저장된 네트워크는 /dli/data/mdt/msvd/video_4096model_iter99 입니다. 파일 이름은 saver.restore(sess, MODEL_NAME) 의 MODEL_NAME이 사용됩니다.

문제:

[1] 수행할 때에만 loss 또는 cost를 계산해야 합니까?

[2] 추론 수행 시, dropout을 사용합니까?

In [ ]:

```
#A npy file of all of the mean feature vectors has been created
val_loader=np.load('/dli/data/mdt/msvd/val_msvd_mean_fc7.npy').tolist()
print('Number of videos in the validation set: ', len(val_loader.keys()))
#Create 2 lists one of the video_id and the other with feature maps.
val_id_key=[]
val_maps_to_id=[]

for observed_vid in val_loader.keys():
        val_id_key.append([observed_vid])
        val_maps_to_id.append(val_loader[observed_vid])

print('number of captions and datapoints',len(val_maps_to_id))
```

In [ ]:

```
tf.reset_default_graph()
batch_size=1
num_steps=20
print_topn=0 #0for do not display
printnumOf=3
#Choose a image to caption
VALDATA=##FIXME## #Val Video Id

variables_dict = {
    "weights_mscoco":tf.Variable(tf.truncated_normal([num_hidden,dict_length],
                                                     stddev=1.0,dtype=tf.float32),name="weights_msc
    "biases_mscoco": tf.Variable(tf.truncated_normal([dict_length],
                                                     stddev=1.0,dtype=tf.float32), name="biases_msc


StartCaption=np.zeros([batch_size,num_steps],dtype=np.int32).tolist()

CaptionPlaceHolder = tf.placeholder(dtype=tf.int32, shape=(batch_size , num_steps))

ValFeatureMap=val_maps_to_id[VALDATA]
X_one_hot=tf.nn.embedding_lookup(np.identity(dict_length), CaptionPlaceHolder) #[batch,num_steps,di
    #ImageFeatureTensor=input_feature_map
Xconcat=tf.concat([ValFeatureMap+tf.zeros([num_steps,batch_size,4096]),
                           tf.unstack(tf.to_float(X_one_hot),num_steps,1)],2)#[:num_steps,:,:]

pred=rnn_model(Xconcat,1.0,1.0,num_layers,num_hidden)
pred=tf.nn.softmax(pred)
saver = tf.train.Saver()

init_op = tf.group(tf.global_variables_initializer(),tf.local_variables_initializer())

with tf.Session() as sess:

    sess.run(init_op)
    coord = tf.train.Coordinator()
    threads = tf.train.start_queue_runners(coord=coord)
    #Load a pretrained network
    saver.restore(sess, '##FIXME##')
    print('Model restored from file')
    for i in range(num_steps-1):
        predict_next_word=sess.run([pred],feed_dict={CaptionPlaceHolder:StartCaption})
        INDEX=np.argmax(predict_next_word[0][i])
        StartCaption[0][i+1]=INDEX
        if print_topn !=0:
            print("Top ",str(printnumOf), "predictions for the", str(i+1), "word in the predicted c
            result_args = np.argsort(predict_next_word[0][i])[-printnumOf:][::-1]
            NextWord=[words[x] for x in result_args]
            print(NextWord)

    coord.request_stop()
    coord.join(threads)
    sess.close()

#print("ground truth caption: ",val_caption_to_id[VALDATA])
v=val_id_key[VALDATA][0]
img=ndimage.imread('/dli/data/mdt/msvd/frames_224/'+v+'/frame_000003_224.bmp')
plt.imshow(img)
```

```
PredictedCaption=[words[x] for x in StartCaption[0]]

print("predicted sentence: ",PredictedCaption[1:])
```

어떻게 하시겠습니까?

여러분은 이미 딥러닝 및 TensorFlow를 사용하여 시각 및 텍스트 데이터를 생성하고 이해하는 방법을 배웠습니다. 이 시점부터 여러분은 성능을 향상 시키기 위해 실습하거나 여러분이 배운 것을 사용할 수 있는 다양한 방법을 연구하실 수 있습니다. 여러분들이 무엇을 해낼지 너무 궁금하네요!

In [ ]:

```
#Free our GPU memory before proceeding to other parts of the lab
import os
os._exit(00)
```

(https://www.nvidia.com/dli)

References

[1] Venugopalan, S., et al. "Translating videos to natural language using deep recurrent neural networks." arXiv preprint arXiv:1412.4729 (2014).

[2] Chen, David L., and William B. Dolan. "Collecting highly parallel data for paraphrase evaluation." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.