I have used editdistance library (which gives Levenshtein distance only) instead of Levenshtein library due to C++ 14.0 errors on my laptop.

In [81]:
```
#!pip install editdistance
```

```
Collecting editdistance
  Downloading https://files.pythonhosted.org/packages/5a/21/3178b32699c94aff68239372e30e01b2707f6b5438d8732d4356162fa3b1/editdistance-0.5.3-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/5a/21/3178b32699c94aff68239372e30e01b2707f6b5438d8732d4356162fa3b1/editdistance-0.5.3-cp37-cp37m-win_amd64.whl)
Installing collected packages: editdistance
Successfully installed editdistance-0.5.3
```

In [90]:
```
# This is the Main function to use Google ASR
# import Levenshtein as Lev
import editdistance
import io
import os
    # Imports the Google Cloud client library
    # [START speech_python_migration_imports]
from google.cloud import speech
from google.cloud.speech import enums
from google.cloud.speech import types
def google_recognize(audio_name,auth_key):

    # [END speech_python_migration_imports]

    os.environ["GOOGLE_APPLICATION_CREDENTIALS"]=auth_key

    # Instantiates a client
    # [START speech_python_migration_client]
    client = speech.SpeechClient()
    # [END speech_python_migration_client]

    # The name of the audio file to transcribe
    file_name = audio_name

    # Loads the audio into memory
    with io.open(file_name, 'rb') as audio_file:
        content = audio_file.read()
        audio = types.RecognitionAudio(content=content)

    config = types.RecognitionConfig(
        encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,
        sample_rate_hertz=16000,
        language_code='en-US')

    # Detects speech in the audio file
    response = client.recognize(config, audio)
    for result in response.results:
#         print(result)
        return result.alternatives[0].transcript
    # [END speech_quickstart]
```

In [91]:
```
def main():
    key = "HW5-DLSP-952006e38a74.json"
    print(google_recognize("S_01_01.wav",key))
main()
```

```
the Birch canoe slid on the smooth planks
```

6. Evaluation matrix

We use the metric of word error rate' (WER) to evaluate the performance of the ASR systems. WER is calculated by first identifying the number of words that the ASR system correctly recognizes. Then the total number of incorrect word substitutions W_S, deletions W_D and

insertions W_I are also counted by comparing the ASR system's output transcription to the ground truth transcription. The value of WER is computed by dividing the sum of substitutions, deletions and insertions by the total number words N_W in the reference transcription.

$$\text{WER} = \frac{W_S + W_D + W_I}{N_W} \times 100$$

We can use Levenshtein distance to calculate the WER between two sentences.

In [92]:
```
#Here is the function calculate WER
def wer(s1, s2):

    s1 =s1.lower()
    s2 =s2.lower()
    b = set(s1.lower().split() + s2.lower().split())

    word2char = dict(zip(b, range(len(b))))


    w1 = [chr(word2char[w]) for w in s1.split()]
    w2 = [chr(word2char[w]) for w in s2.split()]
    return editdistance.eval(''.join(w1), ''.join(w2))/float(len(s2.split()))
```

In [93]:
```
real_sen = "the birch canoe slid on the smooth planks" # Label
key = "HW5-DLSP-952006e38a74.json"
pred_sen = google_recognize("S_01_01.wav",key)
wer_res = wer(pred_sen,real_sen)

print("WER Result:",wer_res )
```

```
WER Result: 0.0
```

Please note that there might be some errors of grpc timeout for requesting to Google API because sometimes internet connectivity drops. In such cases I have resumed previous job from where it left.

In [94]:
```
import os
# test_file_names=[]
# for root, dirs, files in os.walk("TIMIT_full\\TIMIT_full\\test/"):
#     for file in files:
#         if file.endswith(".txt"):
#             test_file_names.append(os.path.join(root, file))
test_txt_files=[]
for root, dirs, files in os.walk("TIMIT_full\\TIMIT_full\\test/"):
    for file in files:
        if file.endswith(".txt"):
            test_txt_files.append(os.path.join(root, file))

test_wav_files=[]
for root, dirs, files in os.walk("TIMIT_full\\TIMIT_full\\test/"):
    for file in files:
        if file.endswith(".wav"):
            test_wav_files.append(os.path.join(root, file))
```

In [95]:
```
# wer_list=[]
for i in range(903,len(test_txt_files),1):
    real_sen=" ".join(open(test_txt_files[i]).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files[i],key)
    wer_res = wer(pred_sen,real_sen)
    wer_list.append(wer_res)
    if(i%50)==0:
        print("At ",i,"WER is : ",wer_res)
```

```
At  0 WER is :  0.09090909090909091
At  50 WER is :  0.18181818181818182
At  100 WER is :  0.09090909090909091
At  150 WER is :  0.09090909090909091
At  200 WER is :  0.18181818181818182
At  250 WER is :  0.09090909090909091
At  300 WER is :  0.09090909090909091
At  350 WER is :  0.18181818181818182
At  400 WER is :  0.18181818181818182
At  450 WER is :  0.18181818181818182
At  500 WER is :  0.18181818181818182
At  550 WER is :  0.18181818181818182
At  600 WER is :  0.09090909090909091
At  650 WER is :  0.18181818181818182
At  700 WER is :  0.18181818181818182
At  750 WER is :  0.18181818181818182
At  800 WER is :  0.09090909090909091
At  850 WER is :  0.09090909090909091
At  900 WER is :  0.09090909090909091
```

In [134]:
```
print(0+len(wer_list))
```

```
902
```

In [108]:
```
wer_list2=[]
for i in range(903,len(test_txt_files),1):
    real_sen=" ".join(open(test_txt_files[i]).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files[i],key)
    wer_res = wer(pred_sen,real_sen)
    wer_list2.append(wer_res)
    if(i%50)==0:
        print("At ",i,"WER is : ",wer_res)
```

```
At  950 WER is :  0.09090909090909091
At  1000 WER is :  0.18181818181818182
At  1050 WER is :  0.18181818181818182
At  1100 WER is :  0.09090909090909091
```

```
---------------------------------------------------------------------------
--
_Rendezvous                               Traceback (most recent call las
t)
C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\grpc_helpers.p
y in error_remapped_callable(*args, **kwargs)
     56         try:
---> 57             return callable_(*args, **kwargs)
     58         except grpc.RpcError as exc:

C:\ProgramData\Anaconda3\lib\site-packages\grpc\_channel.py in __call__(s
elf, request, timeout, metadata, credentials, wait_for_ready, compressio
n)
    689                                       wait_for_ready, compressio
n)
--> 690         return _end_unary_response_blocking(state, call, False, N
one)
    691

C:\ProgramData\Anaconda3\lib\site-packages\grpc\_channel.py in _end_unary
_response_blocking(state, call, with_call, deadline)
    591     else:
--> 592         raise _Rendezvous(state, None, None, deadline)
    593

_Rendezvous: <_Rendezvous of RPC that terminated with:
        status = StatusCode.UNKNOWN
        details = "Stream removed"
        debug_error_string = "{"created":"@1574388467.245000000","descrip
tion":"Error received from peer ipv6:[2607:f8b0:4009:800::200a]:443","fil
e":"src/core/lib/surface/call.cc","file_line":1055,"grpc_message":"Stream
removed","grpc_status":2}"
>

The above exception was the direct cause of the following exception:

Unknown                                   Traceback (most recent call las
t)
<ipython-input-108-36fcdbc21947> in <module>
      2 for i in range(903,len(test_file_names),1):
      3     real_sen=" ".join(open(test_txt_files[i]).read().split(" ")[2
:])
```

```
----> 4     pred_sen = google_recognize(test_wav_files[i],key)
      5     wer_res = wer(pred_sen,real_sen)
      6     wer_list2.append(wer_res)

<ipython-input-90-0e8c9669ffb0> in google_recognize(audio_name, auth_key)
     34
     35     # Detects speech in the audio file
---> 36     response = client.recognize(config, audio)
     37     for result in response.results:
     38 #         print(result)

C:\ProgramData\Anaconda3\lib\site-packages\google\cloud\speech_v1\gapic\speech_client.py in recognize(self, config, audio, retry, timeout, metadata)
    254         request = cloud_speech_pb2.RecognizeRequest(config=config, audio=audio)
    255         return self._inner_api_calls["recognize"](
--> 256             request, retry=retry, timeout=timeout, metadata=metadata
    257         )
    258

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\gapic_v1\method.py in __call__(self, *args, **kwargs)
    141             kwargs["metadata"] = metadata
    142
--> 143         return wrapped_func(*args, **kwargs)
    144
    145

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\retry.py in retry_wrapped_func(*args, **kwargs)
    275                 sleep_generator,
    276                 self._deadline,
--> 277                 on_error=on_error,
    278             )
    279

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\retry.py in retry_target(target, predicate, sleep_generator, deadline, on_error)
    180     for sleep in sleep_generator:
    181         try:
--> 182             return target()
    183
    184         # pylint: disable=broad-except

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\timeout.py in func_with_timeout(*args, **kwargs)
    212             """Wrapped function that adds timeout."""
    213             kwargs["timeout"] = next(timeouts)
--> 214             return func(*args, **kwargs)
    215
    216         return func_with_timeout

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\grpc_helpers.py in error_remapped_callable(*args, **kwargs)
     57             return callable_(*args, **kwargs)
```

```
     58         except grpc.RpcError as exc:
---> 59             six.raise_from(exceptions.from_grpc_error(exc), exc)
     60
     61     return error_remapped_callable

C:\ProgramData\Anaconda3\lib\site-packages\six.py in raise_from(value, fr
om_value)

Unknown: None Stream removed
```

In [137]:
```
print(902+len(wer_list2))
```

```
1143
```

In [119]:
```
wer_list3=[]
for i in range(1143,len(test_txt_files),1):
    real_sen=" ".join(open(test_txt_files[i]).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files[i],key)
    wer_res = wer(pred_sen,real_sen)
    wer_list3.append(wer_res)
    if(i%50)==0:
        print("At ",i,"WER is : ",wer_res)
```

```
At  1150 WER is :  0.36363636363636365
At  1200 WER is :  0.09090909090909091
At  1250 WER is :  0.09090909090909091
At  1300 WER is :  0.09090909090909091
At  1350 WER is :  0.09090909090909091
At  1400 WER is :  0.09090909090909091
At  1450 WER is :  0.18181818181818182
At  1500 WER is :  0.09090909090909091

---------------------------------------------------------------------------
_Rendezvous                               Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\grpc_helpers.py
 in error_remapped_callable(*args, **kwargs)
     56         try:
---> 57             return callable_(*args, **kwargs)
     58         except grpc.RpcError as exc:

C:\ProgramData\Anaconda3\lib\site-packages\grpc\_channel.py in __call__(sel
f, request, timeout, metadata, credentials, wait_for_ready, compression)
    689                                       wait_for_ready, compression)
--> 690         return _end_unary_response_blocking(state, call, False, Non
e)
    691

C:\ProgramData\Anaconda3\lib\site-packages\grpc\_channel.py in _end_unary_r
esponse_blocking(state, call, with_call, deadline)
    591     else:
--> 592         raise _Rendezvous(state, None, None, deadline)
    593

_Rendezvous: <_Rendezvous of RPC that terminated with:
        status = StatusCode.UNKNOWN
        details = "Stream removed"
        debug_error_string = "{"created":"@1574392076.801000000","descripti
on":"Error received from peer ipv6:[2607:f8b0:4009:803::200a]:443","fil
e":"src/core/lib/surface/call.cc","file_line":1055,"grpc_message":"Stream r
emoved","grpc_status":2}"
>

The above exception was the direct cause of the following exception:

Unknown                                   Traceback (most recent call last)
<ipython-input-119-47bd6f67a271> in <module>
      2 for i in range(1143,len(test_txt_files),1):
      3     real_sen=" ".join(open(test_txt_files[i]).read().split(" ")[2:]
)
----> 4     pred_sen = google_recognize(test_wav_files[i],key)
      5     wer_res = wer(pred_sen,real_sen)
```

```
      6     wer_list3.append(wer_res)

<ipython-input-90-0e8c9669ffb0> in google_recognize(audio_name, auth_key)
     34
     35     # Detects speech in the audio file
---> 36     response = client.recognize(config, audio)
     37     for result in response.results:
     38 #         print(result)

C:\ProgramData\Anaconda3\lib\site-packages\google\cloud\speech_v1\gapic\spe
ech_client.py in recognize(self, config, audio, retry, timeout, metadata)
    254         request = cloud_speech_pb2.RecognizeRequest(config=config,
 audio=audio)
    255         return self._inner_api_calls["recognize"](
--> 256             request, retry=retry, timeout=timeout, metadata=metadat
a
    257         )
    258

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\gapic_v1\method.
py in __call__(self, *args, **kwargs)
    141             kwargs["metadata"] = metadata
    142
--> 143         return wrapped_func(*args, **kwargs)
    144
    145

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\retry.py in retr
y_wrapped_func(*args, **kwargs)
    275                 sleep_generator,
    276                 self._deadline,
--> 277                 on_error=on_error,
    278             )
    279

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\retry.py in retr
y_target(target, predicate, sleep_generator, deadline, on_error)
    180     for sleep in sleep_generator:
    181         try:
--> 182             return target()
    183
    184         # pylint: disable=broad-except

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\timeout.py in fu
nc_with_timeout(*args, **kwargs)
    212             """Wrapped function that adds timeout."""
    213             kwargs["timeout"] = next(timeouts)
--> 214             return func(*args, **kwargs)
    215
    216         return func_with_timeout

C:\ProgramData\Anaconda3\lib\site-packages\google\api_core\grpc_helpers.py
 in error_remapped_callable(*args, **kwargs)
     57             return callable_(*args, **kwargs)
     58         except grpc.RpcError as exc:
---> 59             six.raise_from(exceptions.from_grpc_error(exc), exc)
     60
```

```
     61     return error_remapped_callable

C:\ProgramData\Anaconda3\lib\site-packages\six.py in raise_from(value, from
_value)

Unknown: None Stream removed
```

In [133]:
```
print(1143+len(wer_list3))
```

```
1523
```

In [121]:
```
wer_list4=[]
for i in range(1523,len(test_txt_files),1):
    real_sen=" ".join(open(test_txt_files[i]).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files[i],key)
    wer_res = wer(pred_sen,real_sen)
    wer_list4.append(wer_res)
    if(i%50)==0:
        print("At ",i,"WER is : ",wer_res)
```

```
At  1550 WER is :  0.09090909090909091
At  1600 WER is :  0.09090909090909091
At  1650 WER is :  0.09090909090909091
```

In [126]:
```
wer_list_5=wer_list+wer_list2+wer_list3+wer_list4
```

In [127]:
```
len(wer_list_5)
```

Out[127]:
```
1680
```

In [128]:
```
with open('wer_original_list.txt', 'w') as file:
    for item in wer_list_5:
        file.write("%s " % item)
```

Mean and Standard Deviation of WER (Original Data):

In [130]:
```
import numpy as np
print("Mean WER is : ",np.mean(wer_list_5))
print("Std of WER is : ",np.std(wer_list_5))
```

```
Mean WER is :  0.23212161278704016
Std of WER is :  0.1568868017027177
```

In [ ]:
```
os.listdir("PREPARED_DATASET/")[:5]
```

test_wav_files_0, test_wav_files_1, test_wav_files_2, test_wav_files_3 are test wave files that we have prepared

at -5,0,10,25 db SNR levels.

In [141]:
```
test_wav_files_0=[]
for root, dirs, files in os.walk("PREPARED_DATASET/"):
    for file in files:
        if file.endswith("_0.wav"):
             test_wav_files_0.append(os.path.join(root, file))

test_wav_files_1=[]
for root, dirs, files in os.walk("PREPARED_DATASET/"):
    for file in files:
        if file.endswith("_1.wav"):
             test_wav_files_1.append(os.path.join(root, file))

test_wav_files_2=[]
for root, dirs, files in os.walk("PREPARED_DATASET/"):
    for file in files:
        if file.endswith("_2.wav"):
             test_wav_files_2.append(os.path.join(root, file))

test_wav_files_3=[]
for root, dirs, files in os.walk("PREPARED_DATASET/"):
    for file in files:
        if file.endswith("_3.wav"):
             test_wav_files_3.append(os.path.join(root, file))
```

In [162]:
```
ll\\TIMIT_full\\test\\'+'\\'.join(test_wav_files_0[0].split('/')[1].split('__
```

Out[162]: 'TIMIT_full\\TIMIT_full\\test\\dr1\\faks0\\sa1.txt'

In [164]:
```
test_txt_files[0]
```

Out[164]: 'TIMIT_full\\TIMIT_full\\test/dr1\\faks0\\sa1.txt'

In [175]:
```
wer_list_0=[]
for i in range(0,len(test_wav_files_0),1):
    fname='TIMIT_full\\TIMIT_full\\test\\'+'\\'.join(test_wav_files_0[i].spli
    real_sen=" ".join(open(fname).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files_0[i],key)
    if(pred_sen==None):
        pred_sen=''
    wer_res = wer(pred_sen,real_sen)
    wer_list_0.append(wer_res)
    if(i%25)==0:
        print("At file ",i,"WER is : ",wer_res)
```

```
At file  0 WER is :  0.45454545454545453
At file  25 WER is :  2.4
At file  50 WER is :  0.36363636363636365
At file  75 WER is :  0.8571428571428571
At file  100 WER is :  0.45454545454545453
At file  125 WER is :  0.8571428571428571
At file  150 WER is :  0.45454545454545453
At file  175 WER is :  1.5
At file  200 WER is :  0.6363636363636364
At file  225 WER is :  1.4285714285714286
At file  250 WER is :  0.5454545454545454
At file  275 WER is :  1.3333333333333333
At file  300 WER is :  0.45454545454545453
At file  325 WER is :  1.1428571428571428
At file  350 WER is :  0.45454545454545453
At file  375 WER is :  1.0
At file  400 WER is :  0.45454545454545453
At file  425 WER is :  1.0
At file  450 WER is :  0.5454545454545454
At file  475 WER is :  1.375
At file  500 WER is :  0.36363636363636365
At file  525 WER is :  0.875
At file  550 WER is :  0.9090909090909091
At file  575 WER is :  1.0
At file  600 WER is :  0.2727272727272727
At file  625 WER is :  1.0
At file  650 WER is :  0.6363636363636364
At file  675 WER is :  1.8571428571428572
At file  700 WER is :  0.7272727272727273
At file  725 WER is :  0.9
At file  750 WER is :  0.5454545454545454
At file  775 WER is :  1.8333333333333333
At file  800 WER is :  0.36363636363636365
At file  825 WER is :  1.0
At file  850 WER is :  0.2727272727272727
At file  875 WER is :  1.0
At file  900 WER is :  0.36363636363636365
At file  925 WER is :  1.2857142857142858
At file  950 WER is :  0.9090909090909091
At file  975 WER is :  1.125
At file  1000 WER is :  0.36363636363636365
At file  1025 WER is :  1.0
At file  1050 WER is :  0.6363636363636364
```

```
At file  1075 WER is :  1.0
At file  1100 WER is :  0.5454545454545454
At file  1125 WER is :  1.0
At file  1150 WER is :  0.9090909090909091
At file  1175 WER is :  0.8571428571428571
At file  1200 WER is :  0.36363636363636365
At file  1225 WER is :  1.6666666666666667
At file  1250 WER is :  0.18181818181818182
At file  1275 WER is :  1.4285714285714286
At file  1300 WER is :  0.45454545454545453
At file  1325 WER is :  1.1428571428571428
At file  1350 WER is :  0.36363636363636365
At file  1375 WER is :  1.375
At file  1400 WER is :  0.9090909090909091
At file  1425 WER is :  1.0
At file  1450 WER is :  0.45454545454545453
At file  1475 WER is :  1.6666666666666667
At file  1500 WER is :  0.45454545454545453
At file  1525 WER is :  0.875
At file  1550 WER is :  0.6363636363636364
At file  1575 WER is :  1.375
At file  1600 WER is :  0.36363636363636365
At file  1625 WER is :  1.0
At file  1650 WER is :  0.36363636363636365
At file  1675 WER is :  1.125
```

Mean and Standard Deviation of WER for -5 db SNR:

In [177]:

```
print(len(wer_list_0))
print("Mean of WER for -5 db SNR is : ",np.mean(wer_list_0))
print("Std of WER for -5 db SNR is : ",np.std(wer_list_0))
```

```
1680
Mean of WER for -5 db SNR is :  1.0951416702559034
Std of WER for -5 db SNR is :  0.3561457854024702
```

In [181]:
```
wer_list_1=[]
for i in range(0,len(test_wav_files_1),1):
    fname='TIMIT_full\\TIMIT_full\\test\\'+'\\'.join(test_wav_files_1[i].spli
    real_sen=" ".join(open(fname).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files_1[i],key)
    if(pred_sen==None):
        pred_sen=''
    wer_res = wer(pred_sen,real_sen)
    wer_list_1.append(wer_res)
    if(i%25)==0:
        print("At file ",i,"WER is : ",wer_res)
```

```
At file  0 WER is :  0.8181818181818182
At file  25 WER is :  1.0
At file  50 WER is :  0.2727272727272727
At file  75 WER is :  1.0
At file  100 WER is :  0.6363636363636364
At file  125 WER is :  1.2857142857142858
At file  150 WER is :  0.36363636363636365
At file  175 WER is :  0.8333333333333334
At file  200 WER is :  0.5454545454545454
At file  225 WER is :  1.1428571428571428
At file  250 WER is :  0.7272727272727273
At file  275 WER is :  1.1666666666666667
At file  300 WER is :  0.5454545454545454
At file  325 WER is :  1.0
At file  350 WER is :  0.45454545454545453
At file  375 WER is :  0.5714285714285714
At file  400 WER is :  0.6363636363636364
At file  425 WER is :  0.8
At file  450 WER is :  0.45454545454545453
At file  475 WER is :  0.875
At file  500 WER is :  0.45454545454545453
At file  525 WER is :  0.25
At file  550 WER is :  0.7272727272727273
At file  575 WER is :  1.0
At file  600 WER is :  0.2727272727272727
At file  625 WER is :  0.7
At file  650 WER is :  0.2727272727272727
At file  675 WER is :  1.7142857142857142
At file  700 WER is :  0.8181818181818182
At file  725 WER is :  0.9
At file  750 WER is :  0.8181818181818182
At file  775 WER is :  1.6666666666666667
At file  800 WER is :  0.45454545454545453
At file  825 WER is :  1.0
At file  850 WER is :  0.7272727272727273
At file  875 WER is :  0.8333333333333334
At file  900 WER is :  0.09090909090909091
At file  925 WER is :  1.0
At file  950 WER is :  0.9090909090909091
At file  975 WER is :  0.875
At file  1000 WER is :  0.6363636363636364
At file  1025 WER is :  1.0
At file  1050 WER is :  0.8181818181818182
```

```
At file  1075 WER is :  1.2857142857142858
At file  1100 WER is :  0.7272727272727273
At file  1125 WER is :  1.0
At file  1150 WER is :  0.8181818181818182
At file  1175 WER is :  0.7142857142857143
At file  1200 WER is :  0.5454545454545454
At file  1225 WER is :  1.0
At file  1250 WER is :  0.2727272727272727
At file  1275 WER is :  1.0
At file  1300 WER is :  0.5454545454545454
At file  1325 WER is :  0.7142857142857143
At file  1350 WER is :  0.45454545454545453
At file  1375 WER is :  0.75
At file  1400 WER is :  0.7272727272727273
At file  1425 WER is :  0.5714285714285714
At file  1450 WER is :  0.5454545454545454
At file  1475 WER is :  0.6666666666666666
At file  1500 WER is :  0.45454545454545453
At file  1525 WER is :  0.125
At file  1550 WER is :  0.8181818181818182
At file  1575 WER is :  1.0
At file  1600 WER is :  0.2727272727272727
At file  1625 WER is :  1.0
At file  1650 WER is :  0.36363636363636365
At file  1675 WER is :  1.0
```

Mean and Standard Deviation of WER for 0 db SNR:

In [182]:

```
print(len(wer_list_1))
print("Mean of WER for 0 db SNR is : ",np.mean(wer_list_1))
print("Std of WER for 0 db SNR is : ",np.std(wer_list_1))
```

```
1680
Mean of WER for 0 db SNR is :  0.9096140358466209
Std of WER for 0 db SNR is :  0.2860723288963018
```

In [183]:
```
wer_list_2=[]
for i in range(0,len(test_wav_files_2),1):
    fname='TIMIT_full\\TIMIT_full\\test\\'+'\\'.join(test_wav_files_2[i].spli
    real_sen=" ".join(open(fname).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files_2[i],key)
    if(pred_sen==None):
        pred_sen=''
    wer_res = wer(pred_sen,real_sen)
    wer_list_2.append(wer_res)
    if(i%25)==0:
        print("At file ",i,"WER is : ",wer_res)
```

```
At file  0 WER is :  0.36363636363636365
At file  25 WER is :  0.2
At file  50 WER is :  0.18181818181818182
At file  75 WER is :  0.42857142857142855
At file  100 WER is :  0.18181818181818182
At file  125 WER is :  0.7142857142857143
At file  150 WER is :  0.09090909090909091
At file  175 WER is :  0.3333333333333333
At file  200 WER is :  0.2727272727272727
At file  225 WER is :  0.14285714285714285
At file  250 WER is :  0.09090909090909091
At file  275 WER is :  0.8333333333333334
At file  300 WER is :  0.09090909090909091
At file  325 WER is :  0.14285714285714285
At file  350 WER is :  0.18181818181818182
At file  375 WER is :  0.14285714285714285
At file  400 WER is :  0.18181818181818182
At file  425 WER is :  0.1
At file  450 WER is :  0.18181818181818182
At file  475 WER is :  0.125
At file  500 WER is :  0.2727272727272727
At file  525 WER is :  0.125
At file  550 WER is :  0.18181818181818182
At file  575 WER is :  0.14285714285714285
At file  600 WER is :  0.09090909090909091
At file  625 WER is :  0.1
At file  650 WER is :  0.18181818181818182
At file  675 WER is :  0.7142857142857143
At file  700 WER is :  0.09090909090909091
At file  725 WER is :  0.4
At file  750 WER is :  0.2727272727272727
At file  775 WER is :  0.6666666666666666
At file  800 WER is :  0.09090909090909091
At file  825 WER is :  0.125
At file  850 WER is :  0.18181818181818182
At file  875 WER is :  0.5
At file  900 WER is :  0.09090909090909091
At file  925 WER is :  0.7142857142857143
At file  950 WER is :  0.18181818181818182
At file  975 WER is :  0.25
At file  1000 WER is :  0.36363636363636365
At file  1025 WER is :  0.5714285714285714
At file  1050 WER is :  0.2727272727272727
```

```
At file  1075 WER is :  0.8571428571428571
At file  1100 WER is :  0.09090909090909091
At file  1125 WER is :  0.14285714285714285
At file  1150 WER is :  0.36363636363636365
At file  1175 WER is :  0.2857142857142857
At file  1200 WER is :  0.09090909090909091
At file  1225 WER is :  0.3333333333333333
At file  1250 WER is :  0.09090909090909091
At file  1275 WER is :  0.2857142857142857
At file  1300 WER is :  0.18181818181818182
At file  1325 WER is :  0.14285714285714285
At file  1350 WER is :  0.09090909090909091
At file  1375 WER is :  0.125
At file  1400 WER is :  0.09090909090909091
At file  1425 WER is :  0.5714285714285714
At file  1450 WER is :  0.09090909090909091
At file  1475 WER is :  0.16666666666666666
At file  1500 WER is :  0.09090909090909091
At file  1525 WER is :  0.125
At file  1550 WER is :  0.09090909090909091
At file  1575 WER is :  0.125
At file  1600 WER is :  0.09090909090909091
At file  1625 WER is :  0.14285714285714285
At file  1650 WER is :  0.09090909090909091
At file  1675 WER is :  0.25
```

Mean and Standard Deviation of WER for 10 db SNR:

In [184]:

```
print(len(wer_list_2))
print("Mean of WER for 10 db SNR is : ",np.mean(wer_list_2))
print("Std of WER for 10 db SNR is : ",np.std(wer_list_2))
```

```
1680
Mean of WER for 10 db SNR is :  0.3062816113338559
Std of WER for 10 db SNR is :  0.22937985256372115
```

In [185]:
```
wer_list_3=[]
for i in range(0,len(test_wav_files_3),1):
    fname='TIMIT_full\\TIMIT_full\\test\\'+'\\'.join(test_wav_files_3[i].spli
    real_sen=" ".join(open(fname).read().split(" ")[2:])
    pred_sen = google_recognize(test_wav_files_3[i],key)
    if(pred_sen==None):
        pred_sen=''
    wer_res = wer(pred_sen,real_sen)
    wer_list_3.append(wer_res)
    if(i%25)==0:
        print("At file ",i,"WER is : ",wer_res)
```

```
At file  0 WER is :  0.09090909090909091
At file  25 WER is :  0.2
At file  50 WER is :  0.18181818181818182
At file  75 WER is :  0.42857142857142855
At file  100 WER is :  0.09090909090909091
At file  125 WER is :  0.14285714285714285
At file  150 WER is :  0.09090909090909091
At file  175 WER is :  0.3333333333333333
At file  200 WER is :  0.18181818181818182
At file  225 WER is :  0.14285714285714285
At file  250 WER is :  0.09090909090909091
At file  275 WER is :  0.16666666666666666
At file  300 WER is :  0.09090909090909091
At file  325 WER is :  0.14285714285714285
At file  350 WER is :  0.18181818181818182
At file  375 WER is :  0.14285714285714285
At file  400 WER is :  0.18181818181818182
At file  425 WER is :  0.1
At file  450 WER is :  0.18181818181818182
At file  475 WER is :  0.125
At file  500 WER is :  0.18181818181818182
At file  525 WER is :  0.125
At file  550 WER is :  0.18181818181818182
At file  575 WER is :  0.14285714285714285
At file  600 WER is :  0.09090909090909091
At file  625 WER is :  0.1
At file  650 WER is :  0.18181818181818182
At file  675 WER is :  0.14285714285714285
At file  700 WER is :  0.09090909090909091
At file  725 WER is :  0.1
At file  750 WER is :  0.18181818181818182
At file  775 WER is :  0.16666666666666666
At file  800 WER is :  0.09090909090909091
At file  825 WER is :  0.125
At file  850 WER is :  0.18181818181818182
At file  875 WER is :  0.3333333333333333
At file  900 WER is :  0.09090909090909091
At file  925 WER is :  0.5714285714285714
At file  950 WER is :  0.09090909090909091
At file  975 WER is :  0.25
At file  1000 WER is :  0.18181818181818182
At file  1025 WER is :  0.42857142857142855
At file  1050 WER is :  0.18181818181818182
```

```
At file  1075 WER is :  0.2857142857142857
At file  1100 WER is :  0.09090909090909091
At file  1125 WER is :  0.5714285714285714
At file  1150 WER is :  0.36363636363636365
At file  1175 WER is :  0.2857142857142857
At file  1200 WER is :  0.09090909090909091
At file  1225 WER is :  0.3333333333333333
At file  1250 WER is :  0.09090909090909091
At file  1275 WER is :  0.14285714285714285
At file  1300 WER is :  0.18181818181818182
At file  1325 WER is :  0.14285714285714285
At file  1350 WER is :  0.09090909090909091
At file  1375 WER is :  0.125
At file  1400 WER is :  0.09090909090909091
At file  1425 WER is :  0.42857142857142855
At file  1450 WER is :  0.18181818181818182
At file  1475 WER is :  0.16666666666666666
At file  1500 WER is :  0.09090909090909091
At file  1525 WER is :  0.125
At file  1550 WER is :  0.09090909090909091
At file  1575 WER is :  0.125
At file  1600 WER is :  0.09090909090909091
At file  1625 WER is :  0.14285714285714285
At file  1650 WER is :  0.09090909090909091
At file  1675 WER is :  0.25
```

Mean and Standard Deviation of WER for 25 db SNR:

In [186]:
```
print(len(wer_list_3))
print("Mean of WER for 25 db SNR is : ",np.mean(wer_list_3))
print("Std of WER for 25 db SNR is : ",np.std(wer_list_3))
```

```
1680
Mean of WER for 25 db SNR is :  0.2310352824701426
Std of WER for 25 db SNR is :  0.15624296087846523
```

In [ ]:
```

```

Result Summarization

Original Data: Mean WER is : 0.2321 Std of WER is : 0.156

-5db: Mean of WER for -5 db SNR is : 1.095 Std of WER for -5 db SNR is : 0.356

0db:
Mean of WER for 0 db SNR is : 0.909 Std of WER for 0 db SNR is : 0.286

10 db: Mean of WER for 10 db SNR is : 0.306 Std of WER for 10 db SNR is : 0.2293

25 db: Mean of WER for 25 db SNR is : 0.231 Std of WER for 25 db SNR is : 0.1562

Comparison as we increase SNR:

We can clearly see that as we increase SNR, the WER decreases. It is very obvious why this happens as when Signal quality increases, we can see WER to decrease accordingly. Apart from this, we can see that standard deviation also decreases. The reason behind this is that when our quality of signal increases ie when SNR increases, Google API can be more and more confident about predicting speech word because it gets clear speech heard. Now if Google API models confidence increases for each word that it predicts it can lead us to less standard deviation obviously as confidence level for each word increases.

Comparison with Traditional System:

If we compare WER of 0.2321 on original data for Google API with respect to WER of approximately 0.22 from Kaldi, it was astonishing to me at first glance as I really expected google api to perform way more better than Kaldi but it turns out that kaldi performed as well as google API. In fact the Kaldi system performed a bit better when we used 3 Hidden Layers with WER of 21.8.