Kumar Gaurav Singh
B170090CSE

“Finding Similarity Between Talks”

1.ProblemStatement:

According to Oxford dictionary Talk isaconversation or discussion , and we are living in Digital era where all are familiar with blogging, video, Social networks . As an engineering student I am viewing a lot of talks related to technology, Innovation and Entrepreneurship. Its not easy to came up with a totally different talk for a speaker. Somehow they are roamingarounda fixed domain .
My professor Dr. Samya mahuri suggested to find the similarity between the talks.
As by doing this we can measure the tendency that how much the talks of several people are the same (syntactic, semantic).

2.ExistingApproach:

The most common approach we are familiar with at graduate level is :

- Cosine Similarity
- Tf/Idf

Paper “An Effective TF/IDF-based Text-to-Text Semantic Similarity Measure for Text” Classification shows how Tf/Idf can be used to find the semantic similarity between text.

TF-IDF stands for**“Term Frequency — Inverse Document Frequency”**. This is a technique to quantify a word in documents, we generally compute a weight to each word which signifies the importance of the word in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

TF-IDF = Term Frequency (TF) * Inverse Document Frequency (IDF)

3.MyApproach:

Considering Whole talk to evaluate is not a good idea, So we decided to do it on TED Talks video on a particular topic “Artificial Intelligence”.

For the evaluation purpose we choose 5 videos from TED channel that are listed below:

1. Can we build AI without losing control over it by SamHarris.
2. How AI can save our humanity by Kai-FuLee.
3. The danger of AI is weirder than you think by Janelle Shane.
4. The Rise of Artificial intelligence through Deep Learning by YoshuaBengioatTEDxMontreal .
5. Where AI is today and where its going Richard Socher at TEDxSanFrancisco.

Since video can't be used for measuring similarity, So we first Transcribe it by Using a openSource tool “Mozilla Deepspeech”. Which is ASR(Automated Speech Recognition system) in our case. Then we use this transcribe file well known as .txt file for comparing the similarity.

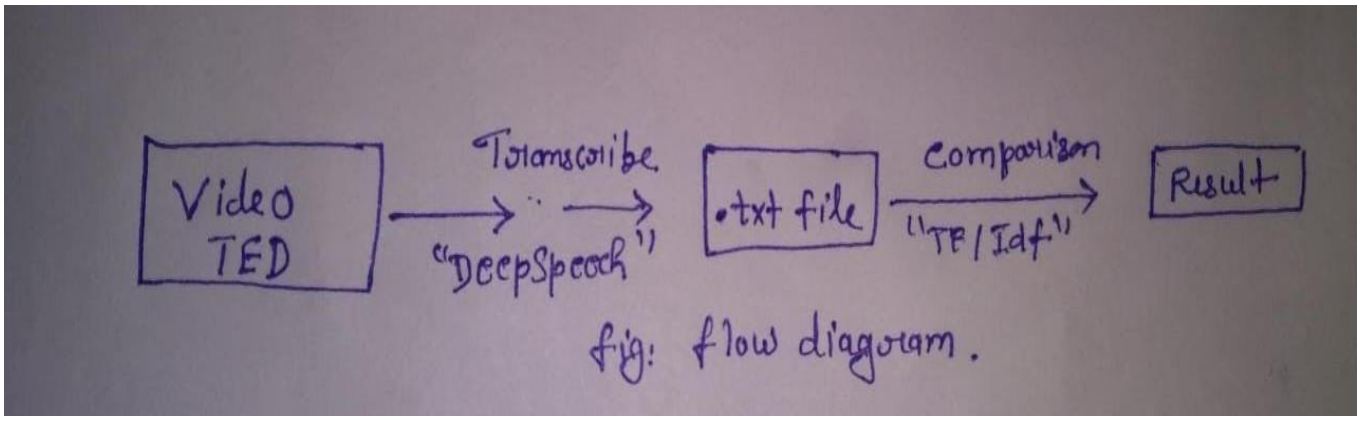

fig: flow diagram.

4.Implementation:

In implementation we have to first prepare the dataset and then proceed to measurethesimilarity.

A.Dataset:

For dataset we have to download all the video mentioned in the [3 My approach] and convert it into audio that will be suitable for Deepspeech. You can use below code to download youtube video and convert it into audio.

```
import pafy
import os
 import subprocess
 import moviepy.editor as mp
url =
"https://www.youtube.com/watch?v=8cmx7V4oIR8"video
= pafy.new(url)

streams = video.streams
for i in streams:
     print(i)
video = pafy.new(url)
k=video.title
print(k)
r=".mp4"
s=".mp4 -ab 160k -ac 2 -ar 44100 -vn audio.wav"
t=".wav"
sdk=k+t
cmd = k+r
print(cmd)
 command="ffmpeg -i How AI can save our humanity | Kai-Fu Lee.mp4 -ab 160k -ac 2 -ar 44100
 -vn audio.wav"
 # get best resolution of a specific format
# set format out of(mp4, webm, flv or 3gp)
best = video.getbest(preftype ="mp4")
best.download()
print("download completed ")
print("its time to convert")
 #subprocess.call(command, shell=True)
 clip = mp.VideoFileClip(cmd).subclip(0,300)
```

clip.audio.write_audiofile(sdk)

After converting to audio use below python script to convert into .txt file

```
Import os
ept = "'deepspeech --model deepspeech-0.6.1-models/output_graph.pbmm --audio
audio1/WhereAIistodayandwhereitsgoingRichardSocherTEDxSanFrancisco.wav"'
ms = os.popen(ept).readlines()
file2 = open("WhereAIistodayandwhereitsgoingRichardSocherTEDxSanFrancisco.txt","w")
file2.writelines(ms)
```

After completing above steps we have data in required format.

B. Tf/Idf :

After getting dataset , Now we have to apply Tf/Idf to compare the similarity
And for comparison we need two txt files , for comparison you can use python Script mentioned below:

```
from sklearn.feature_extraction.text import TfidfVectorizer

text_files = ['../data/nowIsTheTime.txt', '../data/quickBrownFox.txt']
documents = [open(f, encoding="utf8").read() for f in text_files] tfidf
= TfidfVectorizer().fit_transform(documents)
# no need to normalize, since Vectorizer will return normalized tf-idf
pairwise_similarity = (tfidf * tfidf.T).A

print(pairwise_similarity)
```

5. Result andDiscussion:

On running on talks mentioned above in 3[My approach] we get near about more than 60% similarity between them. This shows that Lot of TED speaker speak around 60% same content on same topic . This result signifies that we are inhaling about 60% same content every time. And its also good in different aspects as they are relatively talking good on same topic if wetakeone of them as reference.

6.Conclusion:

Tf/Idf approach is not so good as in last five years of decade we are surprise with the power of deep learning as the amount of data increases. No one is untouched with The RNN weather it is in form of Voice search, Image captioning or other aspects of natural language processing. LSTM, GRU(modified architecture of RNN) are more powerful to measure similarity. I am thankful to Dr. Samya mahuri sir to give me chance to work upon that and continuously guiding me for the same.

7.Reference:

Shereen Albitar, An Effective TF/IDF-based Text-to-Text Semantic Similarity Measure for Text Classification.