CS688 PROJECT
WAV2VEC

SURAAJ SHRESTHA

SPEECH TO TEXT

- WAV2VEC2

  by Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli.

- “Wav2Vec2 is a speech model that accepts a float array corresponding to the raw waveform of the speech signal.”
- Transform Audio wav file of Ted Talk

  by Marco Tempest on "The magic of truth and lies (and iPods)

- Original Text from the Ted Talk Transcript

So the type of magic I like, and I'm a magician, is magic that uses technology to create illusions. So I would like to show you something I've been working on. It's an application that I think will be useful for artists -- multimedia artists in particular. It synchronizes videos across multiple screens of mobile devices. I borrowed these three iPods from people here in the audience to show you what I mean. And I'm going to use them to tell you a little bit about my favorite subject: deception.One of my favorite magicians is Karl Germain. He had this wonderful trick where a rosebush would bloom right in front of your eyes. But it was his production of a butterfly that was the most beautiful. Ladies and gentlemen, the creation of life.

- Transformed Wav2Vec and T-5 Grammer Fixer Model Text

So the type of magic I like and a magician is a magic that uses tecnologis to create illusions so I would like to show you something I've been working on it's an application that I think will be useful for artists. Multimediatistint particular, it synthonizes vivid across multiple screens of mobile devises and I-boro-thy-shee eyepots from people here in the audience to show you what I mean and I-gan a use them to tell you a little bit about my favorite subject (deception). Woo of my favorite magicians is Quarrel German, yet this wonderful trait where a rose bush-woul glow right in front of your eyes. What it was Ti's production of a mildly that was the most beautiful ladies, a gentleman bout creation a like.

```
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
stopwords=list(STOP_WORDS)
from string import punctuation
punctuation=punctuation+ '\n'

nlp = spacy.load('en_core_web_sm')
doc = nlp(sentence)
tokens=[i.text for i in doc]
print(tokens)
```

NLP SUMMARY USING SPACY

Wav2Vec word frequency


```
{'type': 1, 'magic': 2, 'like': 3, 'magician': 1, 'uses': 1, 'tecnologis': 1, 'create': 1, 'illusions': 1, 'working':
1, 'application': 1, 'think': 1, 'useful': 1, 'artists': 1, 'Multimediatistint': 1, 'particular': 1, 'synthonizes':
1, 'vivid': 1, 'multiple': 1, 'screens': 1, 'mobile': 1, 'devises': 1, 'boro': 1, 'thy': 1, 'shee': 1, 'eyepots': 1,
'people': 1, 'audience': 1, 'mean': 1, 'gan': 1, 'use': 1, 'tell': 1, 'little': 1, 'bit': 1, 'favorite': 2, 'subjec
t': 1, 'deception': 1, 'Woo': 1, 'magicians': 1, 'Quarrel': 1, 'German': 1, 'wonderful': 1, 'trait': 1, 'rose': 1, 'b
ush': 1, 'woul': 1, 'glow': 1, 'right': 1, 'eyes': 1, 'Ti': 1, 'production': 1, 'mildly': 1, 'beautiful': 1, 'ladie
s': 1, 'gentleman': 1, 'bout': 1, 'creation': 1}
```


Original word frequency


```
{'type': 1, 'magic': 2, 'like': 2, 'magician': 1, 'uses': 1, 'technology': 1, 'create': 1, 'illusions': 1, 'working':
1, 'application': 1, 'think': 1, 'useful': 1, 'artists': 2, '--': 1, 'multimedia': 1, 'particular': 1, 'synchronize
s': 1, 'videos': 1, 'multiple': 1, 'screens': 1, 'mobile': 1, 'devices': 1, 'borrowed': 1, 'iPods': 1, 'people': 1,
'audience': 1, 'mean': 1, 'going': 1, 'use': 1, 'tell': 1, 'little': 1, 'bit': 1, 'favorite': 2, 'subject': 1, 'decep
tion': 1, 'magicians': 1, 'Karl': 1, 'Germain': 1, 'wonderful': 1, 'trick': 1, 'rosebush': 1, 'bloom': 1, 'right': 1,
'eyes': 1, 'production': 1, 'butterfly': 1, 'beautiful': 1, 'Ladies': 1, 'gentlemen': 1, 'creation': 1, 'life': 1}
```

```
word_frequencies = {}
for i in doc:
    if i.text.lower() not in stopwords:
        if i.text.lower() not in punctuation:
            if i.text not in word_frequencies.keys():
                word_frequencies[i.text] = 1
            else:
                word_frequencies[i.text] += 1

print(word_frequencies)
```

```
maxFreq = max(word_frequencies.values())
for i in word_frequencies.keys():
    word_frequencies[i]=word_frequencies[i]/maxFreq

print(word_frequencies)
```

NLP SUMMARY USING SPACY

Normalized Wav2Vec word frequency


```
{'type': 0.3333333333333333, 'magic': 0.6666666666666666, 'like': 1.0, 'magician': 0.3333333333333333, 'uses': 0.3333
333333333333, 'tecnologis': 0.3333333333333333, 'create': 0.3333333333333333, 'illusions': 0.3333333333333333, 'worki
ng': 0.3333333333333333, 'application': 0.3333333333333333, 'think': 0.3333333333333333, 'useful': 0.333333333333333
3, 'artists': 0.3333333333333333, 'Multimediatistint': 0.3333333333333333, 'particular': 0.3333333333333333, 'synthon
izes': 0.3333333333333333, 'vivid': 0.3333333333333333, 'multiple': 0.3333333333333333, 'screens': 0.333333333333333
3, 'mobile': 0.3333333333333333, 'devises': 0.3333333333333333, 'boro': 0.3333333333333333, 'thy': 0.333333333333333
3, 'shee': 0.3333333333333333, 'eyepots': 0.3333333333333333, 'people': 0.3333333333333333, 'audience': 0.3333333333
33333, 'mean': 0.3333333333333333, 'gan': 0.3333333333333333, 'use': 0.3333333333333333, 'tell': 0.3333333333333333,
'little': 0.3333333333333333, 'bit': 0.3333333333333333, 'favorite': 0.6666666666666666, 'subject': 0.333333333333333
3, 'deception': 0.3333333333333333, 'Woo': 0.3333333333333333, 'magicians': 0.3333333333333333, 'Quarrel': 0.33333333
33333333, 'German': 0.3333333333333333, 'wonderful': 0.3333333333333333, 'trait': 0.3333333333333333, 'rose': 0.33333
3333333333, 'bush': 0.3333333333333333, 'woul': 0.3333333333333333, 'glow': 0.3333333333333333, 'right': 0.333333333
3333333, 'eyes': 0.3333333333333333, 'Ti': 0.3333333333333333, 'production': 0.3333333333333333, 'mildly': 0.33333333
33333333, 'beautiful': 0.3333333333333333, 'ladies': 0.3333333333333333, 'gentleman': 0.3333333333333333, 'bout': 0.3
333333333333333, 'creation': 0.3333333333333333}
```


Normalized Original word frequency


```
{'type': 0.5, 'magic': 1.0, 'like': 1.0, 'magician': 0.5, 'uses': 0.5, 'technology': 0.5, 'create': 0.5, 'illusions':
0.5, 'working': 0.5, 'application': 0.5, 'think': 0.5, 'useful': 0.5, 'artists': 1.0, '--': 0.5, 'multimedia': 0.5,
'particular': 0.5, 'synchronizes': 0.5, 'videos': 0.5, 'multiple': 0.5, 'screens': 0.5, 'mobile': 0.5, 'devices': 0.
5, 'borrowed': 0.5, 'iPods': 0.5, 'people': 0.5, 'audience': 0.5, 'mean': 0.5, 'going': 0.5, 'use': 0.5, 'tell': 0.5,
'little': 0.5, 'bit': 0.5, 'favorite': 1.0, 'subject': 0.5, 'deception': 0.5, 'magicians': 0.5, 'Karl': 0.5, 'Germai
n': 0.5, 'wonderful': 0.5, 'trick': 0.5, 'rosebush': 0.5, 'bloom': 0.5, 'right': 0.5, 'eyes': 0.5, 'production': 0.5,
'butterfly': 0.5, 'beautiful': 0.5, 'Ladies': 0.5, 'gentlemen': 0.5, 'creation': 0.5, 'life': 0.5}
```

```
sentenceScores = {}
for i in sentence_tokens:
    for j in i:
        if j.text.lower() in word_frequencies.keys():
            if i not in sentenceScores.keys():
                sentenceScores[i] = word_frequencies[j.text.lower()]
            else:
                sentenceScores[i] += word_frequencies[j.text.lower()]
print(sentenceScores)
```

NLP SUMMARY USING SPACY

- Wav2Vec text Sentence scores

{So the type of magic I like and a magician is a magic that uses tecnologis to create illusions so I would like to sh
ow you something I've been working on it's an application that I think will be useful for artists.: 6.99999999999999
8, Multimediatistint particular, it synthonizes vivid across multiple screens of mobile devises and I-boro-thy-shee e
yepots from people here in the audience to show you what I mean and I-gan a use them to tell you a little bit about m
y favorite subject (deception).: 7.666666666666664, Woo of my favorite magicians is Quarrel German, yet this wonderfu
l trait where a rose bush-woul glow right in front of your eyes.: 3.666666666666667, What it was Ti's production of a
mildly that was the most beautiful ladies, a gentleman bout creation a like.: 3.333333333333333}

- Original Text Sentence Score

{So the type of magic I like, and I'm a magician, is magic that uses technology to create illusions.: 6.0, So I would
like to show you something I've been working on.: 1.5, It's an application that I think will be useful for artists --
multimedia artists in particular.: 5.0, It synchronizes videos across multiple screens of mobile devices.: 3.0, I bor
rowed these three iPods from people here in the audience to show you what I mean.: 2.0, And I'm going to use them to
tell you a little bit about my favorite subject: deception.: 4.5, One of my favorite magicians is Karl Germain.: 1.5,
He had this wonderful trick where a rosebush would bloom right in front of your eyes.: 3.0, But it was his production
of a butterfly that was the most beautiful.: 1.5, Ladies and gentlemen, the creation of life.: 1.5}

```
from heapq import nlargest
summary1 = nlargest(n = 3 , iterable = sentenceScores , key = sentenceScores.get)
print(summary1)
```

NLP SUMMARY USING SPACY

- Summary of Wav2Vec Text

```
[Multimediatistint particular, it synthonizes vivid across multiple screens of mobile devises and I-boro-thy-shee eye
pots from people here in the audience to show you what I mean and I-gan a use them to tell you a little bit about my
favorite subject (deception)., So the type of magic I like and a magician is a magic that uses tecnologis to create i
llusions so I would like to show you something I've been working on it's an application that I think will be useful f
or artists., Woo of my favorite magicians is Quarrel German, yet this wonderful trait where a rose bush-woul glow rig
ht in front of your eyes.]
```

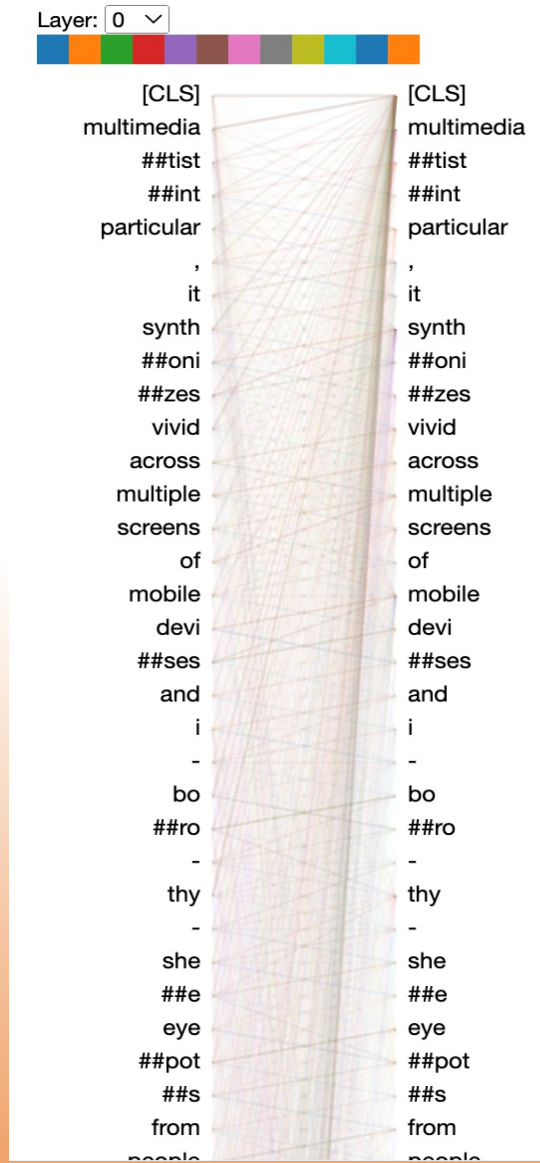
- Summary of the original Text

```
[So the type of magic I like, and I'm a magician, is magic that uses technology to create illusions., It's an applica
tion that I think will be useful for artists -- multimedia artists in particular., And I'm going to use them to tell
you a little bit about my favorite subject: deception.]
```

BERT HEAD VIEW FOR HIGHEST SCORED SENTENCES

```
inputs = tokenizer.encode("Multimediatistint particular, it synthonizes vivid across multiple
outputs = model(inputs)
attention = outputs[-1]  # Output includes attention weights when output_attentions=True
tokens = tokenizer.convert_ids_to_tokens(inputs[0])

from bertviz import head_view
head_view(attention, tokens)
```



CREDITS

- https://huggingface.co/docs/transformers/model_doc/wav2vec2