

Jared De La Serna

Prof. Rob Malouf

LING 571

15 December 2023

Corpus of the WWDC

Introduction

Technology has become a part of our lives and we might not even notice the impact of it. We saw technology improve significantly throughout the years ranging from brick phones to the thinnest phones. As a society, we take technology for granted. We don't understand the numerous ways that it has improved aspects of our lives such as communication, travel, and even security. When we talk about technology, most of the time we associate it with certain companies, certain devices, and certain features. We might associate technology with the company "Apple". In modern day, we can say that Apple has dominated the technology field for the last 15 years. We all remember those days that we would see people outside in the Apple store for days just to get their hands on the television. In their first ever iPhone which was introduced in 2007, it was considered a revolutionary piece of technology, a device that we had never seen before. After 2007, we can safely say that Apple started revolutionising technology each year releasing new software, devices, and technology. We always associate Apple to its founder, Steve Jobs, many people back in the early days saw Steve Jobs as the face of the future, and after Steve Jobs passed away, Tim Cook replaced him as the spokesperson in the WWDC. We can't compare the impact that Steve Jobs had in Apple to Tim Cook's impact. Steve Jobs was a brilliant marketer and is known for his technological advances. People saw him as the "Father of Technology". Every year there is an important meeting called WWDC, hosted for a selective group of students, developers, and influencers. WWDC stands for Worldwide Developers Conference hosted by Apple annually. These meetings are a timestone for technology advancements. They talk about important topics such as operating systems(iOS), new products, app designs, innovation and many more. I decided to compare the WWDC transcripts from various years ranging

from 2007 to 2020. The questions you can ask yourself is why did they change the way that they presented the meetings to the audience? Was it to sell more devices? What made them change the style of speaking of the WWDC? Did they target their meetings to the same audience throughout the years?

Corpus Building

I decided to make the corpus with Python and with the help of some libraries. I imported the packages from the class that followed the same outline, but with a little bit of research I found a class named “Article” from the “newspaper” library. What this library did is basically cleaned the transcript to avoid having ads or certain links in the transcript making our corpus more accurate about the WWDC. I obtained the transcripts from the same source, and I total 10 different transcripts which could give a concrete understanding about the words that they used, the source that I got all the transcripts from the website Singupost, which has most of the year's transcript so obtaining it won't be hard.

```

from newspaper import Article
from nltk import pos_tag, word_tokenize
from urllib.request import urlopen
from urllib.parse import urlsplit
from pathlib import Path
from bs4 import BeautifulSoup
import nltk
import re

all_tokens = []

def get_transcript(url):
    article = Article(url)
    article.download()
    article.parse()
    text = article.text

    data = urlopen(url).read()
    soup = BeautifulSoup(data, 'html.parser')
    content = soup.find('section', class_='body-content')

    tokens = [word_tokenize(sent) for sent in nltk.sent_tokenize(text)]
    tagged = [pos_tag(sent) for sent in tokens]

    folder = Path('singupost')
    filename = folder / (f'{year}_{urlsplit(url).path.replace("/", "_")}.txt')
    folder.mkdir(exist_ok=True)

    with open(filename, 'w') as f:
        for sent in tagged:
            for w, t in sent:
                print(w, t, sep='_', end=' ', file=f)
            print(file=f)

    all_tokens.extend(tokens)

links = [
    'https://singupost.com/full-transcript-tim-cook-at-apple-wwdc-2020-keynote/?singlepage=1',
    'https://singupost.com/full-transcript-tim-cook-at-apple-wwdc-2019-keynote/?singlepage=1',
    'https://singupost.com/tim-cook-at-apple-wwdc-2018-keynote-full-transcript/?singlepage=1',
    'https://singupost.com/apple-ceo-tim-cook-keynote-at-wwdc-2017-full-transcript/?singlepage=1',
    'https://singupost.com/apple-ceo-tim-cook-keynote-at-wwdc-2016-full-transcript/?singlepage=1',
    'https://singupost.com/apple-wwdc-2015-keynote-special-event-june-2015-full-transcript/?singlepage=1',
    'https://singupost.com/apple-ceo-tim-cook-keynote-wwdc-june-2014-transcript/?singlepage=1',
    'https://singupost.com/apple-ceo-tim-cook-keynote-wwdc-june-2013-conference-transcript/?singlepage=1',
    'https://singupost.com/steve-jobs-introduces-iphone-4-facetime-at-wwdc-2010-full-transcript/?singlepage=1',
    'https://singupost.com/steve-jobs-iphone-2007-presentation-full-transcript/?singlepage=1',
]

corpus_folder = Path('singupost')
corpus = nltk.corpus.reader.TaggedCorpusReader(str(corpus_folder), r'[^\s]*\.txt', sep='_')

```

Transcripts are extracted and processed from a list of Apple event URLs using the python followed the outline given in lecture. The `get_transcript` function downloads and parses articles using the newspaper library, uses BeautifulSoup to extract text content, tokenizes words and sentences using NLTK, tags parts of speech, and saves the results to text files. The path from the URL and the year are used to structure filenames. Applying the function to each event URL as iterative going through the list, the script gathers all the tokens into a global list called `all_tokens`. This code, which was created especially for Apple event transcripts, makes it easier to collect and analyse text data systematically from a variety of sources.

```
corpus_folder = Path('singjupost')
corpus = nltk.corpus.reader.TaggedCorpusReader(str(corpus_folder), r'[^.]*\.txt', sep='_')
```

To handle tagged text data, the code generates a `TaggedCorpusReader`, a tool from the NLTK package. It designates a directory ('singjupost') for the storage of tagged text files. The reader is configured to distinguish files that finish in ".txt" and uses underscores to divide tokens which with ending "txt" let us know that is stored as file dataset corpus.. This allows the tagged text data to be easily accessed and analysed within the specified directory in preparation for additional NLTK processing. After I compiled all my code and generated a corpus of all the transcripts of WWDC Apple, I wanted to see how many words were in corpus, so I did the integrated function of Python (`.words()`) and turns out that there is 195338 words in my corpus that could be considered a significant amount.

```
len(corpus.words())
195338
```

MSTR type/token ratio

After this, I was interested in the MSTR for each transcribed year of the WWDC, so I computed the mean standardised type/token ratio (MSTR) for each year. Used a partition size of 1,000 tokens, and normalised the text by removing both punctuation and making it lowercase. We normalised it by using this function below:

```
def normalize(text):
    return [tok.lower() for tok in text if tok.isalpha()]
```

I created a `normalise(text)` function, which outputs a new list with each token transformed to lowercase if it is made up only of alphabetic characters (letters). The MSTR which is a score that is based on the Standardised type/token ratio (MSTR). You just basically divide the text into segments that we get to decide, for this project we did 1000 segments. TTR is basically the type/token ratio which can help us analyse the transcripts, and then we calculate the TTR from each segment and divide it into the mean value of TTRs. The formula is provided down below:

$$\text{type/token ratio} = \frac{\text{no. of types in text or corpus}}{\text{no. of tokens in text or corpus}}$$

We calculated the MSTR by using functions that it was provided in the lectures which was:

```
import nltk
from nltk.tokenize import WhitespaceTokenizer, RegexpTokenizer
import re
from toolz import partition

%precision 3
def normalize(text):
    return [tok.lower() for tok in text if tok.isalpha()]

def ttr(text):
    text = normalize(text)
    return len(set(text)) / len(text)

def mstr(text, k=2000):
    text = normalize(text)
    ttrs = [ttr(chunk) for chunk in partition(k, text)]
    return sum(ttrs) / len(ttrs)
mstr(corpus.words(), k=1000)
```

0.347

We defined 3 functions, MSTR, which determines the MSTR by dividing the text into segments of a given size 2000 and averaging the TTRs of these segments, and TTR, which determines the Type-Token Ratio for a normalised text. Ultimately, a corpus's MSTR is calculated by using `mstr(corpus.words(), k=1000)`, where `k=1000` determines the chunk size. We obtained a MSTR of 0.347, which could be explained that on average there's 34.7% diversity of words per 1000 words in our corpus which could be considered pretty normal. Now if we want to compare it to each other year to see if there is any pattern of the MSTR

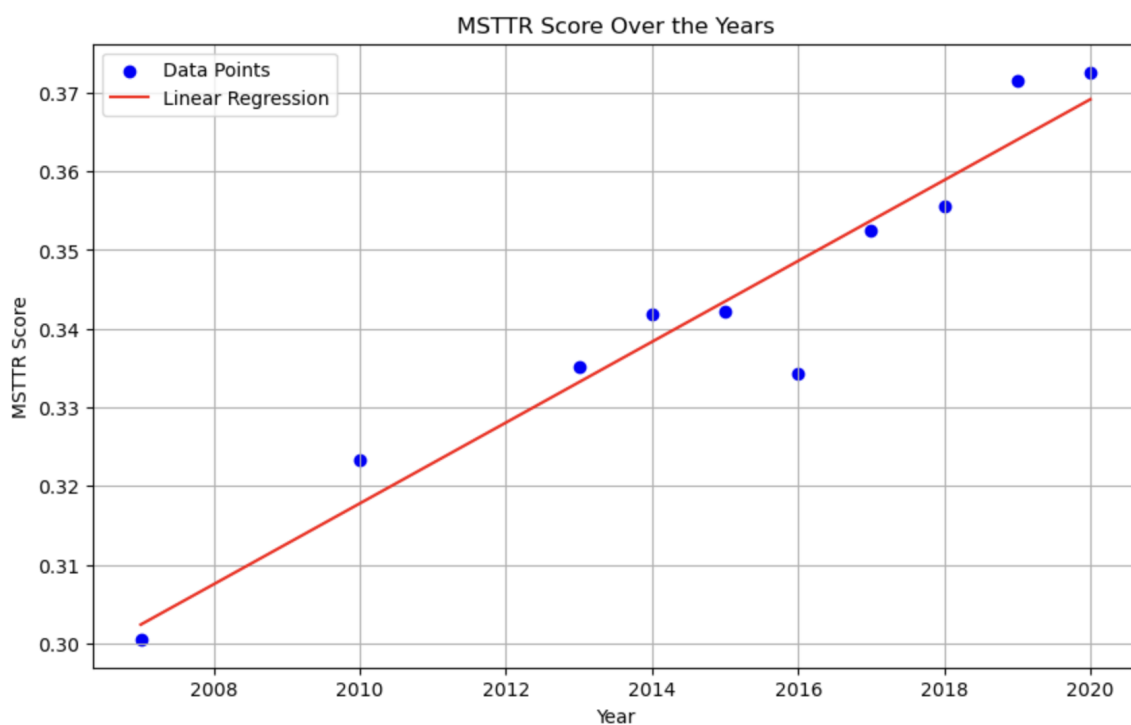
every year which could give a theory that the language that they use changes throughout the years, making us wonder the question why does it change every year and is it marketing strategy or society change. We calculated the MSTR for every year by writing this function below:

```
In [23]: for file_id in corpus.fileids():

    year_match = re.search(r'(\d{4})[^0-9]', file_id)

    if year_match:
        year = year_match.group(1)
        words_in_category = corpus.words(file_id)
        score = mstr(words_in_category, k=1000)
        print(f'Year {year} WWDC MSTR: {score:.3f}')
```

I created a for loop with an if statement integrated that loops over files in a corpus in each file by using a regular expression to extract the year from the file's ID, and prints out the result for each year in the context of the WWDC (Apple Worldwide Developers Conference). Now in order to see the pattern we must graph in a way in order to see them, and we can apply a linear regression to see if the MSTR is going up or down which could help us conclude our theory. Our results indicate that throughout the years the MSTR has significantly increased through the years as we can see in our graph.



The first year of our corpus was 2007 WWDC the score was 0.301 and the most recent transcript of the WWDC was 2020 WWDC was 0.373. We can safely say that throughout the years the diversity in

language of transcripts every year changed. This was an increment of around .72 which could be significant, we can say that the diversity of words in the transcripts throughout the years increases. Which could imply that maybe Apple that people of all ages are becoming more technologically literate, but highlight that younger people are typically more adept with technology.

Flesch-Kincaid Readability Level(FK)

After that we decided to use the Flesch-Kincaid Readability Level (FK), which is basically the calculations to measure the difficulty of every transcript. The formula to calculate the FK we used the following equation:

$$FK = 0.39 \times \frac{\# \text{ of words}}{\# \text{ of sents}} + 11.8 \times \frac{\# \text{ of sylls}}{\# \text{ of words}} - 15.59$$

We defined the function that was given, this was function was used to count the syllabus, the following function was:

```
def syllables(word):
    if word in cmudict:
        return len([p for p in cmudict[word][0] if p[-1].isdigit()])
    else:
        return len(re.findall(r'[aeiou]+', word))
```

The function syllables, defined in this code, determines how many syllables are in a given word. It counts the number of vowels in the word's lowercase form using a regular expression. C mudict is defined such as (cmudict = nltk.corpus.cmudict.dict()) making sure the word is in the cmudict. To calculate the Flesch-Kincaid Readability we used the function that we obtained from the lecture:

```

def fk(text):
    words = len(word_tokenize(text))
    sylls = sum([syllables(w) for w in word_tokenize(text)])
    sents = len(sent_tokenize(text))
    return 0.39 * (words/sents) + 11.8 * (sylls/words) - 15.59

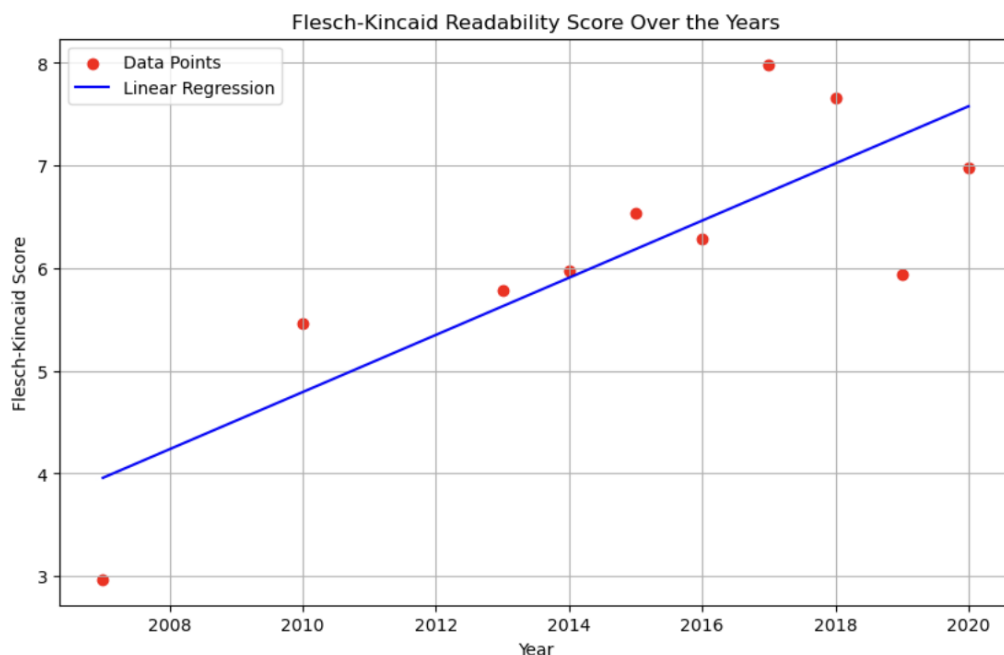
for file_id in project.fileids():

    year_match = re.search(r'(\d{4})[^\d-9]', file_id)

    if year_match:
        year = year_match.group(1)
        full_text = project.raw(file_id)
        score = fk(full_text)
        print(f'Year {year} WWDC FK: {score:.3f}')

```

We basically created a function named “fk” that would compute the formula above between words, syllabus and sents(which is the text tokenized). Then by computing the formula I used an if for loop with an if statement similar to the MTSR situation in order to obtain the FK. After computing the Flesch-Kincaid Readability for every year of WWDC, we can conclude that the score seems to rise quite significantly throughout the years and that throughout the years, the reading level has decreased significantly. I decided to graph them in order to see a correlation. We would compare our (x-variable) which is our year to our FK score which is our (y-variable). The graph came out like this:



I would assume this because we can see that technology is most targeted to our young society, since they are the ones who keep up with these major technology advancements and would be interested in buying devices just to obtain a new feature. In today's society we have babies that could use an iPad better than a senior citizen. I do believe that the WWDC is mostly targeted to young people, people who can take advantage of any technological advancements, maybe the first iPhone was targeted for the mainstream adults.

Collocations

Collocations which are basically expressions of multiple words that happen next to each other. Which can help us understand certain topics, better or maybe suggest a specific meaning. The thing that I wanted to find was trigrams and bigrams, basically bigram is 2 words, and trigrams are 3 words. First we will talk about the bigrams attempting to obtain the top ten trigrams according to their Pointwise Mutual Information (PMI) rankings. Pointwise Mutual Information, or PMI for short, is a statistical metric that quantifies the degree of relationship between two words in a corpus. These are helpful because the “the” “is” simple 2 letters words are ignored. The results are:

```
In [13]: finder3.nbest(trigram_measures.pmi, 10)
```

```
Out[13]: [('Tweet', 'Pinterest', 'Email'),
          ('shares', 'Share', 'Tweet'),
          ('Do', 'Not', 'Disturb'),
          ('Share', 'Tweet', 'Pinterest'),
          ('New', 'York', 'Times'),
          ('Chief', 'Creative', 'Officer'),
          ('third', 'party', 'apps'),
          ('iOS', 'software', 'program'),
          ('Industrial', 'Light', 'and'),
          ('Light', 'and', 'Magic')]
```

We can obtain crucial information with which we would analyse the best trigrams in our transcripts.

We will explain the most important ones that I feel like people will have to understand, first we have “Do”, “Not”, and “Disturb” which is an feature from our iPhone that would stop our phone to make a sound while we receive a notification, a feature that is useful when we are supposed to have our phone

in silent or we are busy. We can also talk about “third”, “party”, “apps” which many people don’t know what it means but we can explain that third apps are an app that comes from a seller who isn’t the company that made the device. These are really important for developers who might develop apps or who are part of the Android community. Now if we talk about “iOS”, “software”, and “program” which is the one of the biggest part of technology that people are most excited about, how their phone software is going to change, this might mean new features to your current phone or even a faster phone or fixes of phone glitches. Now if we want to see the common words are without the PMI, which could be simple words the results are:

```
In [11]: finder.nbest(bigram_measures.likelihood_ratio, 10)
```

```
Out[11]: [('going', 'to'),
          ('you', 'can'),
          ('I', 'can'),
          ('want', 'to'),
          ('of', 'course'),
          ('look', 'at'),
          ('as', 'well'),
          ('App', 'Store'),
          ('d', 'like'),
          ('on', 'the')]
```

The code searches a corpus for and retrieves the top ten bigrams pairs from a corpus. This is really important because even though Steve Jobs and Tim Cook are brilliant sales people they mostly use words such as “you” “can” to make sure the audience feels included about the new technology advances which will secure more sales. They also mostly use “look at” which makes sure to always have the audience's attention at all times in order to make sure they know what they are buying. A surprisingly bigram that showed up was “App” “Store” which is basically a store to download apps for your phone. Every iPhone has experience with the Apple store, which is why they decided to say that a lot every year.

```
In [12]: trigram_measures = nltk.collocations.TrigramAssocMeasures()
finder3 = TrigramCollocationFinder.from_words(tokens)
finder3.apply_word_filter(notalpha)
finder3.apply_freq_filter(10)
finder3.nbest(trigram_measures.likelihood_ratio, 10)
```

```
Out[12]: [('m', 'going', 'to'),
('re', 'going', 'to'),
('going', 'to', 'be'),
('going', 'to', 'make'),
('going', 'to', 'show'),
('going', 'to', 'do'),
('am', 'going', 'to'),
('going', 'to', 'get'),
('going', 'to', 'go'),
('going', 'to', 'bring')]
```

This code, utilising NLTK, distinguishes and positions the best 10 huge trigrams in a text. It selects trigrams that appear at least ten times and applies filters to exclude non-alphabetic words. The positioning depends on the probability proportion measure, underlying factual importance in word affiliations. Most of the words have “going” which I can assume that Steve Jobs and Tim Cook actually use these words just to show the audience that they are “going” to do something that might surprise them. Show something new, that what the words new suggests.

Conclusion

After doing the corpus and taking into account the MSTR, KF, and collocations we can say that their language did change every WWDC meeting. We can solve our questions from the introduction: “What made them change the style of speaking of the WWDC?” We can say that shows a greater frequency of these particular terms, which may imply a change or emphasis in the subjects or information covered in the WWDC. I showed graphs that clearly prove this, also the collocation might show us that they use terms that people who are involved in the Apple community might be aware of. We can safely assume that in the future the MSTR and KF will go up, due to the fact that we will be more aware of the terms that they talk about. The era of technology is barely starting and soon everyone will understand it more. We can finally conclude that technology draws in a younger, more technologically aware audience. Younger people are frequently more eager to discover and accept new features and devices, and they also tend to adjust to technological changes more rapidly, and that younger generations are more adapted to rapid changes in technology which makes them the perfect audience.

Works Cited

1. S, Pangambam. "Apple CEO Tim Cook Keynote at WWDC 2016 (Full Transcript)." *The Singju Post*, 15 June 2016,
<https://singjupost.com/apple-ceo-tim-cook-keynote-at-wwdc-2016-full-transcript/?singlepage=1>.
2. S, Pangambam. "Apple CEO Tim Cook Keynote at WWDC 2017 (Full Transcript)." *The Singju Post*, 7 June 2017,
<https://singjupost.com/apple-ceo-tim-cook-keynote-at-wwdc-2017-full-transcript/?singlepage=1>.
3. S, Pangambam. "Apple CEO Tim Cook Keynote at WWDC June 2013 Conference Transcript." *The Singju Post*, 11 June 2013,
<https://singjupost.com/apple-ceo-tim-cook-keynote-wwdc-june-2013-conference-transcript/?singlepage=1>.
4. S, Pangambam. "Apple CEO Tim Cook Keynote at WWDC June 2014 Transcript." *The Singju Post*, 11 June 2014,
<https://singjupost.com/apple-ceo-tim-cook-keynote-wwdc-june-2014-transcript/?singlepage=1>.
5. S, Pangambam. "Apple WWDC 2015 Keynote – Special Event June 2015." *YouTube*, 30 October 2023,
<https://singjupost.com/apple-wwdc-2015-keynote-special-event-june-2015-full-transcript/?singlepage=1>.
6. S, Pangambam. "Full Transcript: Tim Cook at Apple WWDC 2019 Keynote." *The Singju Post*, 15 June 2019,
<https://singjupost.com/full-transcript-tim-cook-at-apple-wwdc-2019-keynote/?singlepage=1>.
7. S, Pangambam. "Full Transcript: Tim Cook at Apple WWDC 2020 Keynote." *The Singju Post*, 23 June 2020,
<https://singjupost.com/full-transcript-tim-cook-at-apple-wwdc-2020-keynote/?singlepage=1>.

8. S, Pangambam. "Steve Jobs Introduces iPhone 4 & FaceTime at WWDC 2010 (Full Transcript)." *The Singju Post*, 11 February 2017,
<https://singjupost.com/steve-jobs-introduces-iphone-4-facetime-at-wwdc-2010-full-transcript/?singlepage=1>.
9. S, Pangambam. "Steve Jobs iPhone 2007 Presentation (Full Transcript)." *YouTube*, 30 October 2023,
<https://singjupost.com/steve-jobs-iphone-2007-presentation-full-transcript/?singlepage=1>'.
10. S, Pangambam. "Tim Cook at Apple WWDC 2018 Keynote (Full Transcript)." *The Singju Post*, 6 June 2018,
<https://singjupost.com/tim-cook-at-apple-wwdc-2018-keynote-full-transcript/?singlepage=1>.