Cross-Cultural Sentiment Analysis on News Video Descriptions Related to COVID-19

Ting Zou
tz2487@barnard.edu
Advisor: Prof. John R. Kender
Columbia University

Abstract

People all over the world are affected by the coronavirus disease 2019 (COVID-19) pandemic in their daily lives. In general and during lockdown stages, people around the world watched the news to learn the latest information about the pandemic. The number of news videos about COVID-19 has increased on the global video website. The description of the news video can be crucial in this circumstance because most viewers won't watch the entire thing; instead, they'll read the description to understand the major points of the news. Varied cultures could have different preferences for how to phrase the description of the news video. While some may opt to broadcast the news' positive aspects, others may choose to concentrate on its negative aspects. This study aims to examine the disparities in sentiment between the American and Chinese descriptions of the news video.

1. Introduction

The Covid-19 epidemic was initially noted on December 31, 2019, in Wuhan, Hubei Province, China, and it quickly began to spread throughout the world. Finally, on March 11, 2020, as the virus is still spreading, WHO declared the Covid-19 outbreak to be a pandemic. Beginning in China, this virus spread to many other nations, including Italy, Spain, the United States, the United Kingdom, Brazil, and Russia, infecting and killing thousands of people. More than 188 nations and territories reported more than 22.5 million cases of Covid-19 on August 21st, 2020, resulting in more than 7,92,000 fatalities. However, 14.4 million people have been reported as recovering. Millions of people continue to be affected by this pandemic, but many nations have implemented harsh lockdowns for varying lengths of time to break the virus's chain.

The task of categorizing the polarity of a given text is called sentiment analysis. A text-based tweet, for instance, can be classified as "positive," "negative," or "neutral." A model can be trained to predict the right sentiment given the text and related labels.

This study's main goal is to identify cultural differences in the way people from various backgrounds describe news videos. We use the three sentiment categories—"positive," "negative," and "neutral"—to examine the description's emotions. We trained a graph convolutional network on the labeled data to be able to predict the emotion in the provided description and examine the result using the attention score the model produced.

2. Related Work

On the subject of COVID-19, sentiment analysis has been studied extensively. A team of Swiss academics created a model based on BERT to handle sentiment analysis on Twitter about COVID-19. During the COVID-19 outbreak, Twitter has been a useful resource for news and a public forum for expression. However, manually categorizing, filtering, and summarizing the vast quantity of material accessible on COVID-19 on Twitter has proven to be unachievable. Additionally, using technologies from the fields of machine learning and natural language processing to do this work has proven difficult (NLP). Researchers have created a model called COVID-Twitter-BERT to help with the comprehension of Twitter messages relating to COVID-19 material as well as the analysis of this content (CT-BERT). Bidirectional Encoder Representations from Transformers, or BERT, is a brand-new language representation approach. By concurrently conditioning on both left and right context in all layers, BERT is aimed to pre-train deep bidirectional representations from unlabeled text, in contrast to recent language representation models. As a result, without making significant task-specific architecture alterations, the pre-trained BERT model may be improved with just one additional output layer to produce cutting-edge models for a variety of tasks, including question answering and language inference. The CT-BERT model, on the other hand, is trained on a corpus of 160M tweets concerning the coronavirus that were gathered through the Crowdbreaks platform between January 12 and April 16, 2020. Crowdbreaks listens to a list of English-language COVID-19-related keywords2 using the Twitter filter stream API. The original corpus was cleansed for retweet tags before training. All Twitter usernames were changed to a generic text token to pseudonymize each tweet. All URLs to web pages underwent a similar process. After removing all retweets, duplicates, and close duplicates from the dataset, the final corpus consisted of 22.5M tweets, totaling 0.6B words. Therefore, the domain-specific pretraining dataset is 1/7th the size of the dataset used to train the primary base model. Using the spaCy library, tweets were divided into sentences and processed as distinct documents.

Another group of researchers applied Long Short Term Memory(LSTM) Recurrent Neural Network to perform sentiment analysis on Twitter regarding COVID-19 To find sentiments and data labeling, they import Textblob an open-source library in Python built on top of NLTK, that supports complex analysis of data. The Textblob library utilizes the NLTK (Wordnet) interface, a lexical database of English words linked using semantic relationships. The output of the process is an overall polarity score and subjectivity score. The polarity score ranges from '-1' for negative and '+1' for positive sentiments, while the subjectivity score ranges from 0 for objective and 1 for subjective (opinion) sentiments. They calculated the overall sentiment of tweet content as negative, neutral, or positive. Tweets with a score of 1 are labeled as positive and tweets with a score of -1 are labeled as negative. The sentiment label includes extremely positive, extremely negative, negative, neutral, and positive sentiments. The training set for the learning model had 80% of the data, while the test dataset for the model had 20% of the data. They constructed the LSTM- RNN model for this training and testing. Next, they added three layers to the network model including the embedding layer, LSTM layer, and dense layer. Then, they added a dropout layer to help prevent the overfitting of 20% of neurons. The LSTM layers use the input to make predictions to produce an output of predicted values that is close to the actual values.

As sentiment analysis can be seen as a text classification problem, we also researched the paper dealing with text classification. A group of researchers from Northwestern University developed graph convolutional networks for text classification problems. The topic of Graph Neural Networks has received growing attention recently (Cai, Zheng, and Chang 2018; Battaglia et al. 2018). Several authors generalized well-established neural network models like CNN that apply to regular grid structure (2-d mesh or 1-d sequence) to work on arbitrarily structured graphs (Bruna et al. 2014; Henaff, Bruna, and LeCun 2015; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017). In their pioneering work, Kipf and Welling presented a simplified graph neural network model, called graph convolutional networks (GCN), which achieved state-of-the-art classification results on several benchmark graph datasets (Kipf and Welling 2017). GCN was also explored in several NLP tasks such as semantic role labeling (Marcheggiani and Titov 2017), relation classification (Li, Jin, and Luo 2018), and machine translation (Bastings et al. 2017), where GCN is used to encode the syntactic structure of sentences. Some recent studies explored graph neural networks for text classification (Henaff, Bruna, and LeCun 2015; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Peng et al. 2018; Zhang, Liu, and Song 2018). However, they either viewed a document or a sentence as a graph of word nodes (Defferrard, Bresson, and Vandergheynst 2016; Peng et al. 2018; Zhang, Liu, and Song 2018) or relied on the not-routinely-available document citation relation to construct the graph (Kipf and Welling 2017). In contrast, when constructing the corpus graph, the researchers from Northwestern University regard the documents and words as nodes (hence heterogeneous graph) and do not require inter-document relations.

After reviewing these previous works, we found that in the past, although deep learning or neural network methods were also used to extract semantic features when dealing with sentiment analysis problems, the syntactic structural information (actually a kind of graph data) in the text was often ignored. If only the dependency tree is used for rule construction and feature extraction, then the nonlinear semantic relationships among components in the sentence are not learned and exploited. So, we decided to use the dependency tree to build the graph first and then feed the graph into a graph convolutional network.

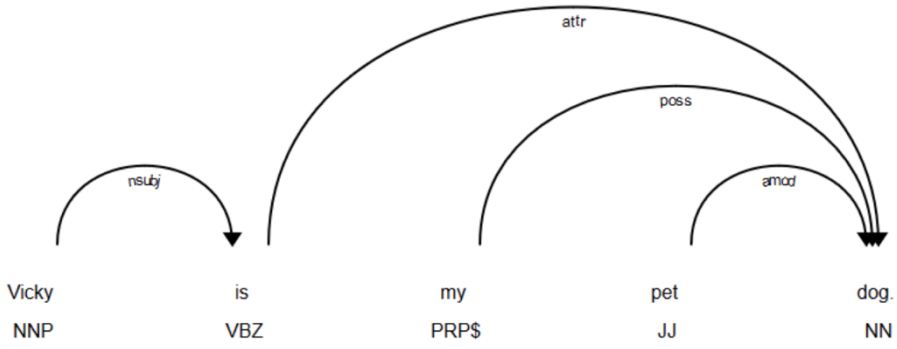

Figure 1 Dependency Tree Example

3. Methods
3.1 Data Collection
3.1.1 Test Dataset

Our initial goal is to recreate the graph convolutional network based on the paper regarding GCN on text classification. For the test dataset, we utilized data "COVIDSenti: A Large-Scale Benchmark Twitter Data Set for COVID-19 Sentiment Analysis**".** It contains four different datasets with sentiment labels on twitter regarding COVID-19.

Dataset\Label	Positive	Negative	Neutral	Total
COVIDSenti-A	1,968	5,083	22,949	30,000
COVIDSenti-B	2,033	5,471	22,496	30,000
COVIDSenti-C	2,279	5,781	21,940	30,000
COVIDSenti	6,280	16,335	67,835	90,000

Figure 2 COVIDSenti Dataset Summary

We utilized the CovidSenti to test the function of the GCN built according to the paper.

COVIDSenti

tweet	label
Coronavirus | Human Coronavirus Types | CDC https://t.co/IxoxVRarIb	neu
@shehryar_taseer That,Äôs true , Corona virus swine flue Bird flu in December when whole Pk is busy in Marriage,Ä¶ https://t.co/6JWBIymnyo	neu
TLDR: Not SARS, possibly new coronavirus. Difficult to confirm because patients identified later in infection when,Ä¶ https://t.co/utKo0fxdgX	neg
Disease outbreak news from the WHO: Middle East respiratory syndrome coronavirus (MERS-CoV) ,Äì The United Arab Emira,Ä¶ https://t.co/n89E94ZILj	neu
China - Media: WSJ says sources tell them mystery pneumonia is a new coronavirus - something that has been speculat,Ä¶ https://t.co/3pJMDuiazw	neu
The mystery new virus causing #China pneumonia outbreak is possibly new coronavirus (same family as #sars and #mers,Ä¶ https://t.co/8OyBQC886H	neu
Virologists weigh in on novel coronavirus in China's outbreak https://t.co/0nx4niHATT	neu
"Chinese authorities have made a preliminary determination of a novel (or new) #coronavirus, identified in a hospit,Ä¶ https://t.co/h0R3XJXEha	neu
@tezuma75 Why #CCP keep on saying unknown cause of pneumonia? The cause is obviously related to corona virus. Let's,Ä¶ https://t.co/wDm1jHmoqX	neg
Chinese report says mysterious illnesses may be from new coronavirus https://t.co/eMnY3pvmJY	neu
China identifies new strain of coronavirus as source of pneumonia outbreak https://t.co/qy1kFhTHDI	neu
I always feel weird hoping for another coronavirus outbreak to rationalize our research!	neg

Figure 3 CovidSenti Example

3.1.2 U.S. News Video Description

We made use of YouTube as the main source of U.S. news videos. We built a web crawler to get the video description from YouTube. For the search keywords, we utilized “covid-19 news 2020”, “covid-19 news 2021”, “covid-19 news 2022”, “covid-19 news

update", and "covid-19 news us". And then we used the selenium package to get the description of each video from the search result.

```
proxies='127.0.0.1:8080'
browser=selenium_api.launch_browser(proxies=proxies)

keyword_list=['covid-19 news 2020','Covid-19 news 2021','Covid-19 news 2022','Covid-19 news update','Covid-19

for keyword in keyword_list:
    print('*'*80)
    print('*'*80)
    search_url='https://www.youtube.com/results?search_query={}'.format(keyword.replace(' ','+'))
    print(search_url)
    browser.get(url=search_url)
    toolkit.sleep(3)
    slide_count=60
    slide_num=0
    while slide_num<slide_count:

        toolkit.print_index(index=slide_num)
        print('keyword',keyword)
        browser.find_element_by_tag_name('body').send_keys(Keys.END)
        toolkit.sleep(3)
        slide_num+=1
```

Figure 4 U.S. News Video Description Web Crawler Code

After getting rid of the repetitive videos, we got around 4k descriptions. As many descriptions include part of advertising like "subscribe to this channel", we only keep the first paragraph of the description which is supposed to be the main body part.

	A	B	C	D
1	title	des	hit	url
2	Federal Rese	Powell will talk about the current state of the economy, the Fed's response to the crisis, and what lies ahead at the Hutchins Center on Fiscal and Monetary Policy at Brookings.	2K	https://www.youtube.co
3	COVID-19: Fe	A CBC News analysis reveals the federal civil service grew by more than 35,000 people over the two-year COVID-19 pandemic — a 12 per cent increase, with four government departments accounting for the lion's share of the new jobs.	20	https://www.youtube.co
4	COVID-19 up	President Biden announced new Covid-19 vaccine incentives including having states use federal funds to award unvaccinated people $100 if they get the vaccine. He also announced new Covid-19 vaccine mandates for federal workers and contractors.	11	https://www.youtube.co
5	Federal heal	More than half of U.S. states have started lifting pandemic restrictions and reopening their economies. But questions remain about how to resume business while maintaining social distancing. In addition, testing for COVID-19 remains relatively limited, with about 250,000 tests per day conducted nationwide. Judy Woodruff talks to Dr. Ashish Jha, director of the Harvard Global Health Institute.	38	https://www.youtube.co
6	Federal Gove	The White House on Thursday finalized its vaccine mandate for businesses nationwide with over 100 employees.	15	https://www.youtube.co

Figure 5 U.S. News Video Description Demo

For the data labeling part, we manually labeled the 4k news video descriptions with the label “positive”, “negative”, and “neutral”. In order to make the labeling process fast, the “positive” tag is marked with 1, “negative” is marked with 2, and “neutral” is marked with 0.

3.1.3 Chinese News Video Description

For the Chinese news video description, we used bilibili as the source for the new videos. We built another web crawler for bilibili using search keywords “新冠疫情新闻 2020”, “新冠疫情新闻 2021”, “新冠疫情新闻 2022”, “新冠疫情新闻更新”, “新冠疫情新闻中国”, that is the Chinese version of “covid-19 news 2020”, “covid-19 news 2021”, “covid-19 news 2022”, “covid-19 news update”, and “covid-19 news China”.

```
def get_video_detail(bvid,headers=None,proxies=None,verify=True):
    """
    视频的信息\n
    视频的名称、描述、时长等基本信息
    """
    api='https://www.bilibili.com/video/{}'.format(bvid)
    response=toolkit.get(url=api,headers=headers,proxies=proxies,timeout=120,verify=verify)
    response.encoding='utf-8'
    # print(response.text)

    # 视频、音频链接
    playinfo_str=re.findall('__playinfo__=(.*?)</script>',response.text,re.S)[0]
    playinfo=json.loads(playinfo_str)
    # 音频
    audio=playinfo['data']['dash']['audio']
    audio_url=audio[0]['baseUrl']
    audio_bandwidth=audio[0]['bandwidth']
    # 视频
    video=playinfo['data']['dash']['video']
    video_url=video[0]['baseUrl']
    video_width=video[0]['width']

    # 其他信息
    state_str=re.findall('__INITIAL_STATE__=(.*?);',response.text,re.S)[0]
```

Figure 6 Chinese New Video Description Web Crawler Code

For the Chinese news video description, after getting rid of the duplicate videos, we got around 2k descriptions. Due to time reasons, we did not label the Chinese news video

钟南山：新出现的新冠病毒变异毒株IHU，新变异毒株IHU的变异位点比奥密克戎多，有46个，奥密克戎是35个，从理论上它对疫苗的免疫逃逸能力，以及对一些抗体的抵抗力可能会强一些。但对于新的变异毒株，还有很多是未知的，现在需要进行比较严密的观察，一个是它的传播力会不会成为主流，另一个是它的致病率是否比较轻。西安此轮疫情的拐点应该已经出现，新增确诊一直在往下降。（总台央视记者 陈旭婷 宋雪 陈杰雄 陈恒杰）	//i1.hdslb.com/bfs/archive
香港疫情仍然严重，在没可能清零的情况下，政府将在4月底放宽社交距离措施，开放西方国家航班，香港未来何去何从?	//i1.hdslb.com/bfs/archive
中国新冠疫苗究竟有多厉害？在国际上的表现令世界惊叹！	//i2.hdslb.com/bfs/archive
5月14日，中国疾控中心研究员、科研攻关组疫苗研发专班专家组成员邵一鸣表示，我国的灭活疫苗对新冠病毒变异株仍有保护效果。	//i1.hdslb.com/bfs/archive
来了！中国新冠病毒疫苗已获批上市	//i0.hdslb.com/bfs/archive
今天，国家卫健委副主任、国务院联防联控机制科研攻关组疫苗研发专班负责人曾益新介绍，新冠病毒疫苗的基本属性是公共产品，价格根据使用规模的大小会有所变化，但是一个大前提肯定是为全民免费提供。（总台央视记者余静英）	//i2.hdslb.com/bfs/archive

3.2 Method

3.2.1 Graph Convolutional Networks(GCN)

A GCN is a multilayer neural network that operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods. Formally, consider a graph $G = (V, E)$, where $V(|V| = n)$ and E are sets of nodes and edges, respectively. Every node is assumed to be connected to itself, i.e., $(v, v) \in E$ for any v. Let $X \in \mathbb{R}^{n \times m}$ be a matrix containing all n nodes with their features, where m is the dimension of the feature vectors, each row $x_v \in \mathbb{R}^m$ is the feature vector for v. We introduce an adjacency matrix A of G and its degree matrix D, where $D_{ii} = \sum_j A_{ij}$. The diagonal elements of A are set to 1 because of self-loops. GCN can capture information only about immediate neighbors with one layer of convolution. When multiple GCN layers are stacked, information about larger neighborhoods are integrated. For a one-layer GCN, the new k-dimensional node feature matrix $L^{(1)} \in \mathbb{R}^{n \times k}$ is computed as

$$L^{(1)} = \rho\left(\tilde{A}XW_0\right)$$

where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix and $W_0 \in \mathbb{R}^{m \times k}$ is a weight matrix. ρ is an activation function, e.g. a ReLU $\rho(x) = \max(0, x)$. As mentioned before, one can incorporate higher order neighborhoods information by stacking multiple GCN layers:

$$L^{(j+1)} = \rho\left(\tilde{A}L^{(j)}W_j\right)$$

where j denotes the layer number and $L^{(0)} = X$.

3.2.2 Text Graph Convolutional Networks

We build a large and heterogeneous text graph which contains word nodes and document nodes so that global word co-occurrence can be explicitly modeled and graph convolution can be easily adapted. The number of nodes in the text graph $|V|$ is the number of documents (corpus size) plus the number of unique words (vocabulary size) in a corpus. We simply set feature matrix $X = I$ as an identity matrix which means every word or document is represented as a one-hot vector as the input to Text GCN. We build edges among nodes based on word occurrence in documents (document-word edges) and word co-occurrence in the whole corpus (word-word edges). The weight of the edge between a document node and a word node is the term frequency-inverse document frequency (TF-IDF) of the word in the document, where term frequency is the number of times the word appears in the document, inverse document frequency is the logarithmically scaled inverse fraction of the number of documents that contain the word. We found using TF-IDF weight is better than using term frequency only. To utilize global word co-occurrence information, we use a fixed size sliding window on all documents in the corpus to gather co-occurrence statistics. We employ point-wise mutual information (PMI), a popular measure for word associations, to calculate weights between two word nodes. We also found using PMI achieves better results than using word co-occurrence count in our preliminary experiments. Formally, the weight of edge between node i and node j is defined as

$$A_{ij} = \begin{cases} \text{PMI}\ (i,j) & i,j \text{ are words, PMI}\ (i,j) > 0 \\ \text{TF} - IDF_{ij} & i \text{ is document, } j \text{ is word} \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

The PMI value of a word pair i, j is computed as

$$\begin{aligned} \text{PMI}\ (i,j) &= \log \frac{p(i,j)}{p(i)p(j)} \\ p(i,j) &= \frac{\#W(i,j)}{\#W} \\ p(i) &= \frac{\#W(i)}{\#W} \end{aligned}$$

where $\#W(i)$ is the number of sliding windows in a corpus that contain word i, $\#W(i,j)$ is the number of sliding windows that contain both word i and j, and $\#W$ is the total number of sliding windows in the corpus. A positive PMI value implies a high semantic correlation of words in a corpus, while a negative PMI value indicates little or no semantic correlation in the corpus. Therefore, we only add edges between word pairs with positive PMI values.

After building the text graph, we feed the graph into a simple two layer GCN, the second layer node (word/document) embeddings have the same size as the labels set and are fed into a softmax classifier:

$$Z = \text{softmax}\left(\tilde{A}\text{ReLU}\left(\tilde{A}XW_0\right)W_1\right)$$

where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the same as before, and $\text{softmax}(x_i) = \frac{1}{\mathcal{Z}}\exp(x_i)$ with $\mathcal{Z} = \sum_i \exp(x_i)$. The loss function is defined as the cross-entropy error over all labeled documents:

$$\mathcal{L} = -\sum_{d \in \mathcal{Y}_D}\sum_{f=1}^{F} Y_{df} \ln Z_{df}$$

where $\mathcal{Y}_D$ is the set of document indices that have labels and F is the dimension of the output features, which is equal to the number of classes. Y is the label indicator matrix. The weight parameters W_0 and W_1 can be trained via gradient descent. In equation 7, $E_1 = \tilde{A}XW_0$ contains the first layer document and word embeddings and $E_2 = \tilde{A}\text{ReLU}\left(\tilde{A}XW_0\right)W_1$ contains the second layer document and word embeddings. The overall Text GCN model is schematically illustrated in Figure 8.

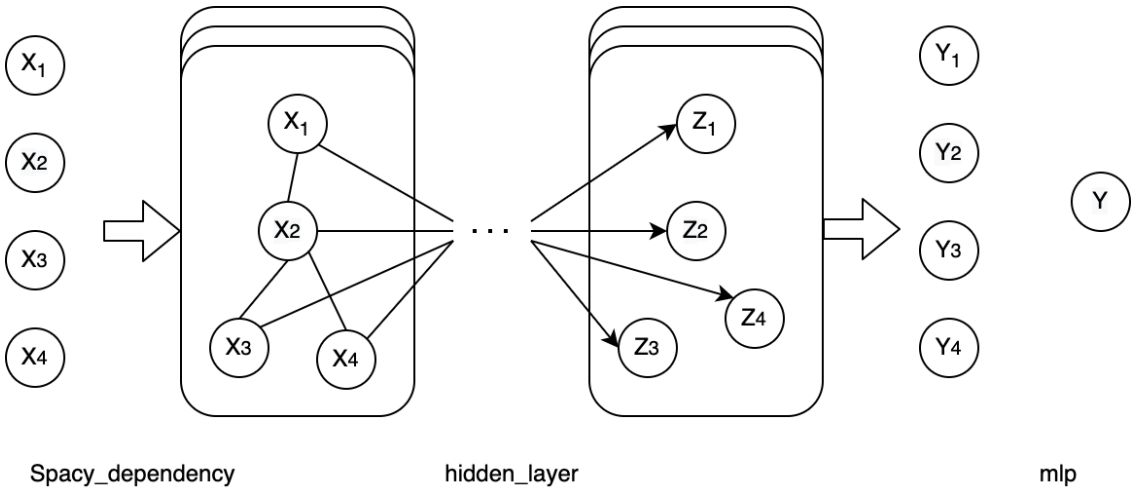

Figure 8 Model Structure

4. Results

4.1 Initial Exploration

Our first step is to test the function of the GCN built according to the formula in the paper. We utilized the CovidSenti dataset for this initial exploration. The first step is to use the Spacy package to tokenize twitter from CovidSenti and build the graph for GCN based on the dependency tree. Then, we generate the adjacency matrix based on the graph to feed into the model. After that, we feed the adjacency matrix into the two-layer GCN model.

```
def dependency_parsing(text_list, pad_size):
    seq = text_list
    seq_len = len(seq)
    text = ' '.join(seq)
    doc = nlp(text)
    token_idx = 0
    token2seq = {}
    for seq_idx in range(seq_len):
        for _ in nlp(seq[seq_idx]):
            token2seq[token_idx] = seq_idx
            token_idx += 1
    origin_adj = [[0.0] * pad_size for _ in range(pad_size)]
    try:
        for token in doc:
            origin_adj[token2seq[token.i]][token2seq[token.i]] = 1.0
            for child in token.children:
                origin_adj[token2seq[token.i]][token2seq[child.i]] = 1.0
                origin_adj[token2seq[child.i]][token2seq[token.i]] = 1.0
        print('true')
    except Exception as error:
        # print(text)
        # print([d for d in doc])
        # print(error)
        print('error')
    return origin_adj
```

Figure 9 Spacy & Adjacency Matrix Code

```python
def gcn(self, x, adj, first_layer=False):
    # gcn
    denom = adj.sum(2).unsqueeze(2) + 1  # norm adj
    Ax = torch.matmul(adj, x)
    AxW = self.W(Ax)
    AxW = AxW / denom
    gAxW = F.relu(AxW)
    out = gAxW if first_layer else self.drop(gAxW)
    return out

def forward(self, x):
    x, _, adj = x
    out = self.embedding(x)  # [batch_size, seq_len, embeding]=[128, 32, 300]

    out = self.gcn(out, adj, first_layer=True)

    for _ in range(1, self.num_layers):
        out = self.gcn(out, adj, first_layer=False)

    out = F.relu(self.fc1(out))  # 句子最后时刻的 hidden state
    out = out.sum(dim=1)  # ave pooling
    out = self.fc(out)  # 句子最后时刻的 hidden state
    return out
```

Figure 10 GCN Model Code

After 500 epochs the train accuracy was 100% and the validation accuracy was 67.29%. This showed the model's structure was successfully built and could be used for further exploration.

```
Iter:  17600,  Train Loss: 0.0095,  Train Acc: 99.22%,  Val Loss:   7.6,  Val Acc: 67.29%,
Epoch [490/500]
Epoch [491/500]
Epoch [492/500]
Epoch [493/500]
Epoch [494/500]
Epoch [495/500]
Iter:  17800,  Train Loss: 0.00015,  Train Acc: 100.00%,  Val Loss:   7.8,  Val Acc: 67.29%,
Epoch [496/500]
Epoch [497/500]
Epoch [498/500]
Epoch [499/500]
Epoch [500/500]
```

Figure 11 CovidSenti Training Result

4.2 U.S. and Chinese News Video Description

Then we used the labeled U.S. news video description dataset to train the model. The training accuracy was 100% and the validation accuracy was 93.64%. The GCN model was performing well on the sentiment analysis task regarding COVID-19

```
Epoch [487/500]
Iter:  14600,  Train Loss:  0.17,  Train Acc: 95.31%,  Val Loss:  0.42,  Val Acc: 93.01%,  Time: 0:05:23
Epoch [488/500]
Epoch [489/500]
Epoch [490/500]
Epoch [491/500]
Epoch [492/500]
Epoch [493/500]
Epoch [494/500]
Iter:  14800,  Train Loss: 0.033,  Train Acc: 100.00%,  Val Loss:   0.4,  Val Acc: 93.64%,  Time: 0:05:27
Epoch [495/500]
Epoch [496/500]
Epoch [497/500]
Epoch [498/500]
Epoch [499/500]
Epoch [500/500]
```

Figure 12 U.S. News Video Description Training Result

For the Chinese news video description dataset, as we do not have enough time to label it, we first utilized google translate API to translate the description into English. Then we use the model trained on the U.S. news video description to make a prediction on the sentiment for the translated Chinese description.

4.3 Attention Score

We added an attention layer to the original GCN, so each word and the relation between two words in the sentence is assigned by an attention score. The higher the attention score, the more weight the word or relation contributes to the final decision of the label.

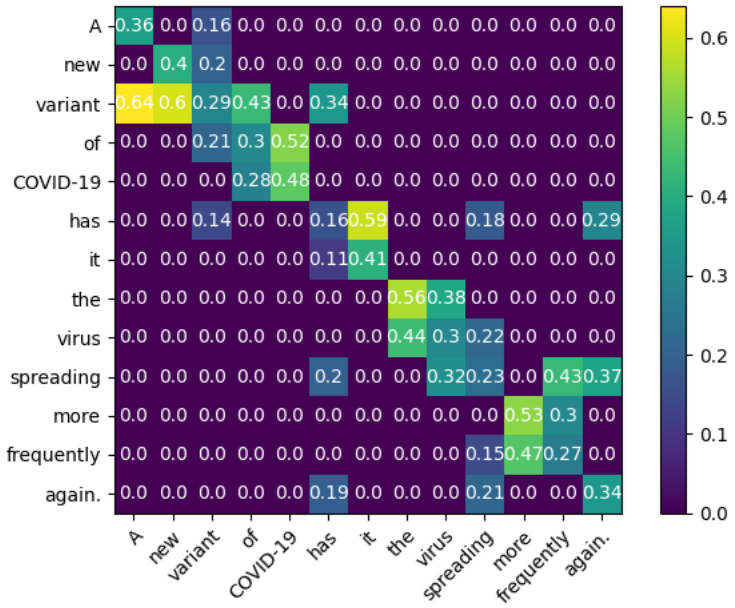

Figure 13 Attention Score Heat Map Example

We extracted the attention score for each word from both U.S. and Chinese datasets and sort the word by the average attention score it received from the model. We separated the word by the label ‘positive’ and ‘negative’, hoping to find the cultural difference between the word used.

```
df=pd.DataFrame(attention_positve)
display(df.sort_values(by=['avg_attention_score'],ascending=False))
```

	word	avg_attention_score
0	favorite	0.67
1	decrease	0.65
2	subscribe	0.52
3	postive	0.51
4	support	0.49
5	help	0.41
6	recent	0.38
7	announce	0.37
8	update	0.36
9	effective	0.32

Figure 14 U.S. Positive Description Word Attention Score

```
df=pd.DataFrame(attention_negative)
display(df.sort_values(by=['avg_attention_score'],ascending=False))
```

	word	avg_attention_score
0	headline	0.63
1	forget	0.62
2	criticize	0.58
3	FoxNews	0.52
4	Biden	0.49
5	form	0.48
6	nothing	0.37
7	shutdown	0.37
8	case	0.35
9	vaccine	0.34

Figure 15 U.S. Negative Description Word Attention Score

```
df=pd.DataFrame(attention_positive)
display(df.sort_values(by=['avg_attention_score'],ascending=False))
```

	word	avg_attention_score
0	new	0.45
1	the	0.42
2	on	0.41
3	it	0.39
4	this	0.34
5	at	0.34
6	first	0.31
7	portal	0.29
8	crown	0.28
9	how	0.28

Figure 16 Chinese Positive Description Word Attention Score

```
df=pd.DataFrame(attention_negative)
display(df.sort_values(by=['avg_attention_score'],ascending=False))
```

	word	avg_attention_score
0	new	0.48
1	the	0.47
2	on	0.47
3	it	0.39
4	this	0.36
5	at	0.35
6	first	0.31
7	portal	0.28
8	crown	0.28
9	how	0.28

Figure 17 Chinese Negative Description Word Attention Score

We can see from the U.S. positive attention score that the top-ranked words are associated with positive emotion. And surprisingly, for the U.S. negative attention score, President Biden's name is ranked as the top word with a high attention score. We suppose this is because President Biden may be criticized for his policy regarding COVID-19 a lot in the news video.

On the other hand, the attention score result for the Chinese description is not ideal. We guessed it was because the translation is bad. So, we investigated the translation version of the Chinese description and found out the translation is indeed a bad one. The google translate API translated the Chinese word separately by its character. For example, "新冠", the Chinese word for COVID-19, was translated into "new crown", and "new" means "新" while "crown" means "冠". In order to find the appropriate word sorted by attention score for Chinese description, we tried another translation called DeepL. However, this API makes the same mistake as google translate.

4.4 Graph Comparison

To find out why the model fails on the Chinese news video description, we extracted the dependency parsing tree from the model and confirmed the reason indeed is the translation. The adjacency matrix that feeds in the GCN is based on the dependency tree. If the tree generated by the Spacy package is wrong, the result can't be accurate.

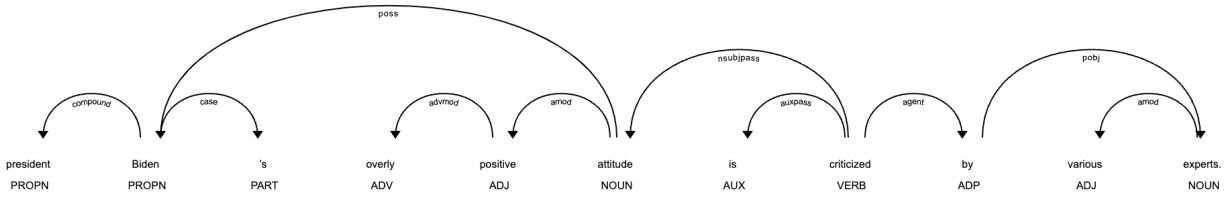

Figure 18 U.S. Description Dependency Tree Example

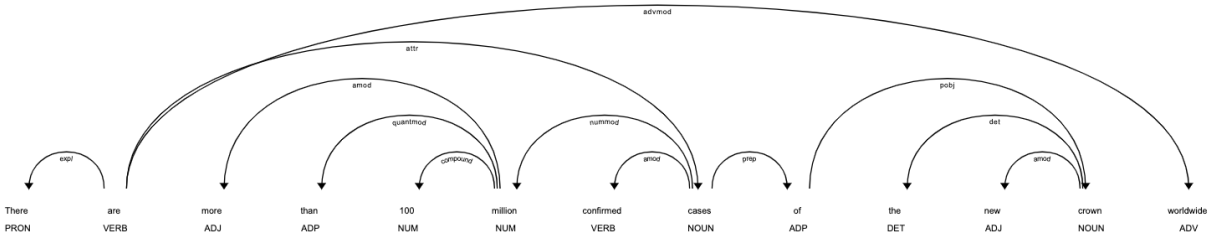

Figure 19 Translated Chinese Description Dependency Tree Example

From the figure above, we can see that for the translated Chinese description, "new crown", the wrong translation for "COVID-19", is treated as an adjective and a noun. This probably is the reason that "new" and "crown" receives high attention score as shown in the figure in section 4.3.

5. Future Work

From the result, we can see that there is some interesting discovery in the U.S. news video descriptions. It's a topic worth exploring, so we may collect and label more data for the U.S. descriptions. We may do more analysis on the labeled data such as word frequency analysis to dig deeper into the way U.S. people narrate the video description.

For the Chinese description part, we should first label the description, and then find a way to tokenize and build the graph in the Chinese version. Then we could train a model from the Chinese dataset. After that, we could compare the word with a high attention score from both U.S. and China. We may discover some differences between the descriptions of the two cultures.

6. Reference

[1] Li, Q.; Han, Z.; and Wu, X. 2018. Deeper insights into graph convolutional networks for semisupervised learning. In AAAI.

[2] Li, Y.; Jin, R.; and Luo, Y. 2018. Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seggcrns).
Journal of the American Medical Informatics Association DOI: 10.1093/jamia/ocy157

[3] Liang Yao, Chengsheng Mao, Yuan Luo. Graph Convolutional Networks for Text Classification.

[4] Ghaida Alorini, Danda B. Rawat. LSTM-RNN Based Sentiment Analysis to Monitor COVID-19 Opinions using Social Media Data.

[5] Martin Muller, Marcel Salathe, Per E Kumervold. Covid-Twitter-Bert: A Natural Language Processing Model to Analyse Covid-19 Content on Twitter.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018