Allison Donine

20 April 2021

POLS/PPUA 7346: Resilient Cities

Northeastern University

Project Title: Who Responds First? The Resilience of Women-Focused Organizations to COVID-19 in Metro Manila

The goals of this paper are to:

(1) Identify the gaps women-focused organizations (WFOs) serve in responding to vulnerable groups' needs and their response to COVID-19 in Metro Manila, (2) investigate to what extent the Philippine Government has prioritized investments to support women-focused organizations' or the needs of women and other vulnerable groups, and (3) connect the work of WFOs to city-wide long-term recovery and resilience.

Methods:

In addition to in-depth interviews with female leaders of WFOs active in Metro manila, I conducted an in-depth text-analysis of every resolution published by the Philippines COVID-19 Inter-Agency Task Force for the Management of Emerging Infectious Diseases (IATF-EID) between March 2020 and May 2020. The IATF-EID was first convened in January 2020 and on March 25, revealed a National Action Plan (NAP). The plan was established to implement and decentralize the system of managing the virus in the Philippines. It is chaired by the Department of Health and includes membership from all executive departments spanning environmental, defense, labor, social welfare, transportation, education, agriculture, and more. The intention of this analysis was to examine the priorities of the Philippine Government's response to COVID-19 and the extent to which a gender and social lens was deployed in the decision-making process.

Text-Analysis of Government Documents:

Using Python, I collected and analyzed the text of 33 resolutions spanning the first three months of the COVID-19 pandemic in the Philippines.

```
In [230]: with open ("/Users/allisondonine/Dropbox/My Mac (Allison's MacBook Air)/
          Desktop/IATF_Resolutions/IATF_RESO_9_to_41.txt", 'r', encoding = 'utf-8'
          ) as myfile1:
              reso_9_41 = myfile1.read()
```

```
In [231]: import string
          PUNCT_LIST = string.punctuation
          STOP_WORDS = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselve
          s', 'you', 'your', 'yours',
                        'yourself', 'yourselves', 'he', 'him', 'his', 'hims
          elf', 'she', 'her', 'hers', 'herself',
                        'it', 'its', 'itself', 'they', 'them', 'their', 'th
          eirs', 'themselves', 'what', 'which',
                        'who', 'whom', 'this', 'that', 'these', 'those', 'a
          m', 'is', 'are', 'was', 'were', 'be',
                        'been', 'being', 'have', 'has', 'had', 'having', 'd
          o', 'does', 'did', 'doing', 'a', 'an',
                        'the', 'and', 'but', 'if', 'or', 'because', 'as',
          'until', 'while', 'of', 'at', 'by', 'for',
                        'with', 'about', 'against', 'between', 'into', 'thr
          ough', 'during', 'before', 'after',
                        'above', 'below', 'to', 'from', 'up', 'down', 'in',
          'out', 'on', 'off', 'over',
                        'under', 'again', 'further', 'then', 'once', 'here'
          , 'there', 'when', 'where', 'why',
                        'how', 'all', 'any', 'both', 'each', 'few', 'more',
          'most', 'other', 'some', 'such', 'no',
                        'nor', 'not', 'only', 'own', 'same', 'so', 'than',
          'too', 'very', 'can', 'will',
                        'just', 'dont', 'should', 'aint', 'arent', 'couldn'
          , 'could', 'would', 'much', 'must',
                        'didnt', 'doesnt', 'hadnt', 'hasnt', 'havent', 'isn
          t', 'mightnt', 'mustnt', 'neednt', 'shan',
                        'shouldnt', 'wasnt', 'werent', 'wont', 'wouldnt']
```

I cleaned and tokenized the text. Then created a function for producing word frequnecies.

In [232]:
```python
import string
PUNCT_LIST= string.punctuation

def tokenize_text(text):
    text = text.lower()
    text_clean = ''.join([e for e in text if e not in PUNCT_LIST])
    text_token =  text_clean.split()
    text_token_clean = [word for word in text_token if word not in STOP_
WORDS]
    return text_token_clean

def count_words(word_list):
    counts_dict = dict()
    for word in word_list:
        if word not in counts_dict:
            counts_dict[word] = 1
        else:
            counts_dict[word] += 1
    return counts_dict

def sorted_counts(word_frequency_dict):
    freq_words = []
    for key, val in word_frequency_dict.items():
        freq_words.append((val, key))
    freq_words.sort(reverse = True)
    return(freq_words)
```

In [233]:
```python
word_list_9_41 = tokenize_text(reso_9_41)
word_frequency_dict_9_41 = count_words(word_list_9_41)
sorted_counts_9_41 = sorted_counts(word_frequency_dict_9_41)
```

In [234]:
```python
sorted_counts_9_41[:5]
```

Out[234]:
```
[(284, '2020'),
 (171, 'whereas'),
 (150, 'covid19'),
 (132, 'quarantine'),
 (125, 'iatf')]
```

In [235]:
```python
def collect_word_frequency(word_frequency_dict, word):
    word_count = sum(
        [word_frequency_dict[key]
            for key in word_frequency_dict.keys()
            if word in key]
    )
    return(word, word_count)

collect_word_frequency(word_frequency_dict_9_41, 'women')
```

Out[235]: ('women', 0)

In [236]:
```python
print(word_frequency_dict_9_41['trade'])
```

```
1
```

```python
In [237]: def total_word_frequency(word_frequency_dicts, word):
              total = 0
              for word_frequency_dict in word_frequency_dicts:
                  total = total + collect_word_frequency(word_frequency_dict, word
          )[1]
              return(word, total)


          total_word_frequency([word_frequency_dict_9_41], 'test')

Out[237]: ('test', 7)
```

I built the program by first compiling a list of words associated with four themes: gender mainstreaming, social amelioration, health & safety, and economic recovery. I then developed a function to compare the frequency of terms associated with each theme across all 33 resolutions.

```python
In [238]: target_words_gender_mainstreaming = ['women', 'gender', 'kasarian' 'equa
          lity', 'pagkakapantay-pantay', 'buntis' 'pregnant women', 'women-led',
                                               'babae', 'mga bata', 'reproductive'
          , 'woman', 'gender-sensitive', 'gender mainstreaming', 'gender-based']
          target_words_social_amelioration =  ['social', 'workers', 'harship', 'ch
          ildren', 'families', 'assistance',
                                               'financial assistance','family', 'p
          amilya','relief', 'kaluwagan', 'social welfare',
                                               'hardships', 'DSWD', 'Overseas Fili
          pino Workers', "OFWs", 'homeless', 'farmers','subsidy',
                                               'senior citizens' , 'beneficiaries'
          ,'poverty','poor', 'mahirap',
                                               'unequal burden', 'burden', 'stress
          es', 'assistance']
          target_words_economic_recovery = ['economy', 'economic', 'trade', 'econo
          mic recovery', 'GDP']
          target_words_testing_vaccine = ['testing', 'test', 'vaccine', 'health',
          'quarantine']
          target_words_political = ['terorism', 'terroris', 'activist', 'terorist
          a' 'conflcit', 'drugs','communist', 'communism']


          def target_word_frequency(word_frequency_dict, target_words):
              total = 0
              for word in target_words:
                  total = total + total_word_frequency([word_frequency_dict], word
          )[1] #calling into one dictionary
              return(total)
```

```python
In [239]: # gender target words:
          target_word_frequency(word_frequency_dict_9_41, target_words_gender_main
          streaming)

Out[239]: 0
```

In [240]:
```
# social target words:
target_word_frequency(word_frequency_dict_9_41, target_words_social_amel
ioration)
```

Out[240]: 48

In [241]:
```
# Health target words:
target_word_frequency(word_frequency_dict_9_41, target_words_testing_vac
cine)
```

Out[241]: 226

In [242]:
```
#economic target words:
target_word_frequency(word_frequency_dict_9_41, target_words_economic_re
covery)
```

Out[242]: 17

In [243]:
```
# political target words:
target_word_frequency(word_frequency_dict_9_41, target_words_political)
```

Out[243]: 0

In [244]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [246]:
```
df = pd.DataFrame({
    'Gender Mainstreaming (0)': [0],
    'Social Amelioration':[48], 'Health': [226], 'Economic Recovery': [1
7],
    }, index = [''])
df
```

Out[246]:

	Gender Mainstreaming (0)	Social Amelioration	Health	Economic Recovery
	0	48	226	17

```
In [253]:  df.plot(kind = 'bar')
           plt.legend(loc = 10, bbox_to_anchor = (1.35, .92))
           plt.title('IATF Resolutions: March- May 2020')
           plt.ylabel('Word Frequency')
           plt.show()
```
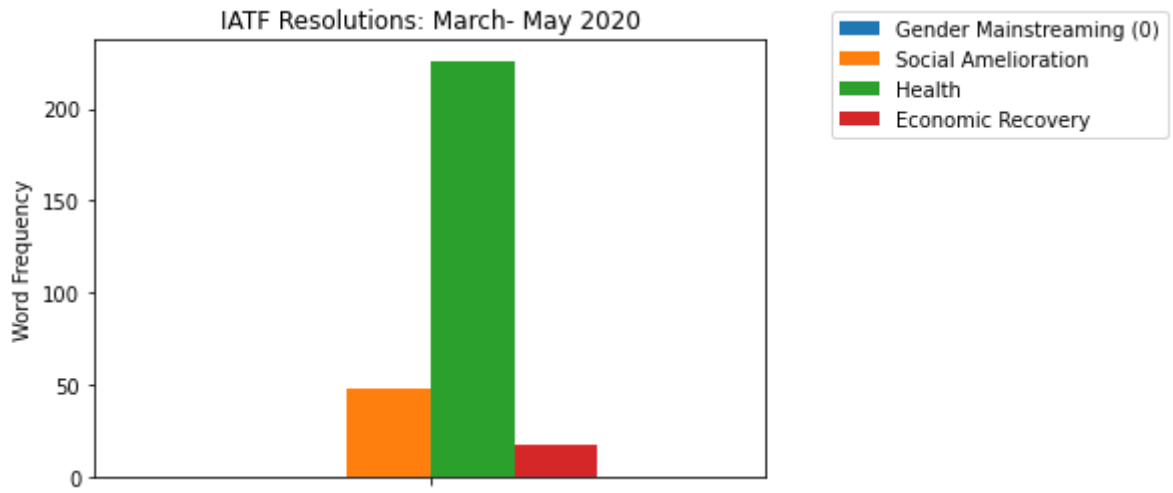


```
In [254]:  plt.savefig('IATF_Resolutions.png')
```

```
<Figure size 432x288 with 0 Axes>
```

As the bar chart demonstrates, no terms associated with gender mainstreaming (i.e. 'women,' 'gender,' 'equality,' 'women-led,' etc.) were identified across the 33 resolutions and only a few terms associated with social amelioration were located in the text. The majority of resolutions centered on discussions of health and economic recovery. These findings further validate concerns raised by feminist activists and WFOs throughout the country.