In [1]:
```
# importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import re
import numpy as np
```

In [2]:
```
#1
#read data set

meta_data = pd.read_csv('/Users/zhiyi/Desktop/540/metadata.csv')
print('Original Size of Data:',meta_data.shape)

#drop rows with null values (based on abstract attribute)
meta_data.dropna(subset = ['abstract'],axis = 0, inplace = True)
print('Data Size after dropping rows with null values (based on abstract attribute):',meta_data.shape)
```

```
/Users/zhiyi/opt/anaconda3/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3369: DtypeWarning: Columns (1,4,5,6,13,14,15,16) have mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
Original Size of Data: (1056660, 19)
Data Size after dropping rows with null values (based on abstract attribute): (821118, 19)
```

In [3]:
```
#handling duplicate data (based on 'sha','title' and 'abstract')
print(meta_data[meta_data.duplicated(subset=['sha','title','abstract'], keep=False) == True])
meta_data.drop_duplicates(subset=['sha','title','abstract'],keep ='last',inplace=True)
print('Data Size after dropping duplicated data (based on abstract attribute):',meta_data.shape)
```

```
        cord_uid  sha                source_x  \
7528    6r981q0t  NaN                     PMC
7529    8qcd85x7  NaN                     PMC
10554   j3mb9zr4  NaN                     PMC
10710   zhw8vh3e  NaN                     PMC
10805   smm5s0ai  NaN                     PMC
...          ...  ...                     ...
1055884  lob7rary  NaN             Medline; PMC
1056362  ejprabi5  NaN             Medline; PMC
1056463  h37h7tqm  NaN  Elsevier; Medline; PMC
1056511  g5vqg0k8  NaN             Medline; PMC
1056586  65doyfvd  NaN             Medline; PMC

                                                    title  \
7528     Management in Ausnahmesituationen: Taktisches ...
7529     Management in Ausnahmesituationen: Taktisches ...
10554    Infektionsschutzrecht nach Inkrafttreten des M...
10710    Artificial Intelligence (AI) applications for ...
10805    Infektionsschutzrecht nach Inkrafttreten des M...
...                                                    ...
1055884  Neuropilin-1 facilitates SARS-CoV-2 cell entry...
1056362  Digital Pathology During the COVID-19 Outbreak...
1056463  The countries that tamed covid-19
1056511  Programming course for health science as a str...
1056586  Social Media Use for Health Purposes: Systemat...

                                  doi       pmcid   pubmed_id     license  \
7528          10.1007/s00740-011-0347-2  PMC7111714    32288861      no-cc
7529          10.1007/s00735-011-0469-1  PMC7111728    32288313      no-cc
10554         10.1007/s00120-020-01212-x  PMC7184239    32338303      no-cc
10710           10.1016/j.dsx.2020.04.012  PMC7195043    32305024      no-cc
10805     10.1007/s00129-020-04606-2  PMC7203504    32382165      no-cc
...                                  ...         ...         ...         ...
1055884          10.1126/science.abd2985  PMC7857391  33082293.0      cc-by
1056362                  10.2196/24266  PMC7901595  33503002.0      cc-by
1056463  10.1016/s0262-4079(20)32220-x  PMC7833330  33518990.0  els-covid
1056511        10.1152/advan.00183.2020  PMC8083174  33464193.0      no-cc
1056586                  10.2196/17917  PMC8156131  33978589.0      cc-by

                                                  abstract publish_time  \
7528     Im öffentlichen Gesundheitsdienst und vor alle...   2011-04-27
7529     Im öffentlichen Gesundheitsdienst und vor alle...   2011-07-19
10554    On 1 March 2020, the amendments to the German ...   2020-04-27
10710    BACKGROUND AND AIMS: Healthcare delivery requi...   2020-04-14
10805    On 1 March 2020, the amendments to the German ...   2020-05-07
...                                                    ...          ...
1055884  The causative agent of coronavirus disease 201...   2020-11-13
1056362  BACKGROUND: Transition to digital pathology us...   2021-02-22
1056463  A handful of nations have achieved something t...   2020-12-26
1056511  Programming is an important skill for differen...   2021-03-01
1056586  BACKGROUND: Social media has been widely used ...   2021-05-12

                                                   authors  \
7528                                          Seidl, Franz
7529                                            Seidl, F.
10554                                       Lissel, P. M.
10710    Vaishya, Raju; Javaid, Mohd; Khan, Ibrahim Hal...
10805                                       Lissel, P. M.
...                                                    ...
1055884  Cantuti-Castelvetri, Ludovico; Ojha, Ravi; Ped...
1056362  Giaretto, Simone; Renne, Salvatore Lorenzo; Ra...
1056463                                    Le Page, Michael
1056511  De la Fuente, Carlos I.; Guadagnin, Eliane Cel...
1056586                                 Chen, Junhan; Wang, Yuan

                journal  mag_id who_covidence_id arxiv_id  \
7528       Wien Klin Mag     NaN             NaN      NaN
7529             Procare     NaN             NaN      NaN
10554         Urologe A      NaN             NaN      NaN
10710   Diabetes Metab Syndr     NaN             NaN      NaN
10805          Gynakologe     NaN             NaN      NaN
...                   ...     ...             ...      ...
1055884            Science     NaN             NaN      NaN
1056362  J Med Internet Res     NaN             NaN      NaN
1056463       New Scientist     NaN             NaN      NaN
1056511     Adv Physiol Educ     NaN             NaN      NaN
1056586  J Med Internet Res     NaN             NaN      NaN

        pdf_json_files                                pmc_json_files  \
7528               NaN                                           NaN
7529               NaN                                           NaN
10554              NaN                                           NaN
10710              NaN  document_parses/pmc_json/PMC7195043.xml.json
10805              NaN                                           NaN
...                ...                                           ...
1055884            NaN  document_parses/pmc_json/PMC7857391.xml.json
1056362            NaN  document_parses/pmc_json/PMC7901595.xml.json
1056463            NaN                                           NaN
1056511            NaN  document_parses/pmc_json/PMC8083174.xml.json
1056586            NaN  document_parses/pmc_json/PMC8156131.xml.json

                                                    url        s2_id
7528     https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7...         NaN
7529     https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7...         NaN
10554    https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7...         NaN
10710    https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7...         NaN
10805    https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7...         NaN
...                                                    ...         ...
1055884  https://doi.org/10.1126/science.abd2985; https...  224823992.0
1056362  https://www.ncbi.nlm.nih.gov/pubmed/33503002/;...  231762843.0
1056463  https://www.sciencedirect.com/science/article/...  231705623.0
1056511  https://www.ncbi.nlm.nih.gov/pubmed/33464193/;...  231641109.0
1056586  https://www.ncbi.nlm.nih.gov/pubmed/33978589/;...  234471309.0

[77329 rows x 19 columns]
Data Size after dropping duplicated data (based on abstract attribute): (780393, 19)
```

In [4]:
```
#3
#function to deal with null values
#'No Information Available' will be replaced
def dealing_with_null_values(dataset):
    dataset = dataset
    for i in dataset.columns:
        replace = []
        data  = dataset[i].isnull()
        count = 0
        for j,k in zip(data,dataset[i]):
            if (j==True):
                count = count+1
                replace.append('No Information Available')
            else:
                replace.append(k)
        print("Num of null values :",i,":",count)
        dataset[i] = replace
    return dataset

meta_data = dealing_with_null_values(meta_data)
```

```
Num of null values : cord_uid : 0
Num of null values : sha : 453723
Num of null values : source_x : 0
Num of null values : title : 101
Num of null values : doi : 247393
Num of null values : pmcid : 468365
Num of null values : pubmed_id : 376968
Num of null values : license : 0
Num of null values : abstract : 0
Num of null values : publish_time : 1723
Num of null values : authors : 4697
Num of null values : journal : 76604
Num of null values : mag_id : 780393
Num of null values : who_covidence_id : 460430
Num of null values : arxiv_id : 766222
Num of null values : pdf_json_files : 453723
Num of null values : pmc_json_files : 509070
Num of null values : url : 221696
Num of null values : s2_id : 55458
```

In [5]:
```
meta_data.shape
```

Out[5]:
```
(780393, 19)
```

In [6]:
```
from sklearn.decomposition import PCA
def pca_fun(n_components, data):
    pca = PCA(n_components=n_components).fit(data)
    data = pca.transform(data)
    return data
```

In [7]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

def tfidf(data):
    tfidf = TfidfVectorizer( stop_words='english',use_idf=True)
    tfidf_matrix = tfidf.fit_transform(data)
    return tfidf_matrix
```

In [8]:
```
# Let's create a matrix with tfidf for the column abstract
tfidf_matrix = tfidf(meta_data['abstract'])
```

In [9]:
```
# in order to explore which documents have more similar respresentaiton, consine simliartiy can be used
from sklearn.metrics.pairwise import linear_kernel
cosine_similarities = linear_kernel(tfidf_matrix[0:1], tfidf_matrix).flatten()

# 10 most related documents indices
related_docs_indices = cosine_similarities.argsort()[:-11:-1]
print("Related Document:",related_docs_indices)

# Cosine Similarties of related documents
print("Cosine Similarities of related documents",cosine_similarities[related_docs_indices])
```

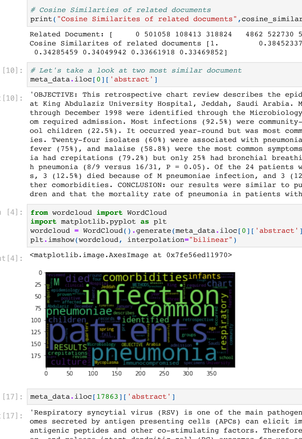
```
Related Document: [     0 501058 108413 318824   4862 522730 581986 471695 128297 625805]
Cosine Similarites of related documents [1.         0.38452337 0.35538984 0.35166117 0.34838092 0.34520779
 0.34285459 0.34049942 0.33661918 0.33469852]
```

In [10]:
```
# Let's take a look at two most similar document
meta_data.iloc[0]['abstract']
```

Out[10]:
```
'OBJECTIVE: This retrospective chart review describes the epidemiology and clinical features of 40 patients with culture-proven Mycoplasma pneumoniae infections at King Abdulaziz University Hospital, Jeddah, Saudi Arabia. METHODS: Patients with positive M. pneumoniae cultures from respiratory specimens from January 1997 through December 1998 were identified through the Microbiology records. Charts of patients were reviewed. RESULTS: 40 patients were identified, 33 (82.5%) of whom required admission. Most infections (92.5%) were community-acquired. The infection affected all age groups but was most common in infants (32.5%) and pre-school children (22.5%). It occurred year-round but was most common in the fall (35%) and spring (30%). More than three-quarters of patients (77.5%) had comorbidities. Twenty-four isolates (60%) were associated with pneumonia, 14 (35%) with upper respiratory tract infections, and 2 (5%) with bronchiolitis. Cough (82.5%), fever (75%), and malaise (58.8%) were the most common symptoms, and crepitations (60%), and wheezes (40%) were the most common signs. Most patients with pneumonia had crepitations (79.2%) but only 25% had bronchial breathing. Immunocompromised patients were more likely than non-immunocompromised patients to present with pneumonia (8/9 versus 16/31, P = 0.05). Of the 24 patients with pneumonia, 14 (58.3%) had uneventful recovery, 4 (16.7%) recovered following some complications, 3 (12.5%) died because of M pneumoniae infection, and 3 (12.5%) died due to underlying comorbidities. The 3 patients who died of M pneumoniae pneumonia had other comorbidities. CONCLUSION: our results were similar to published data except for the finding that infections were more common in infants and preschool children and that the mortality rate of pneumonia in patients with comorbidities was high.'
```

In [4]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
wordcloud = WordCloud().generate(meta_data.iloc[0]['abstract'])
plt.imshow(wordcloud, interpolation='bilinear')
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x7fe56ed11970>
```

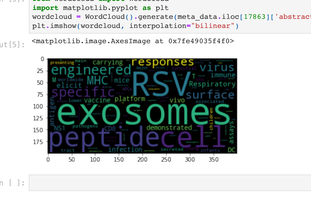


In [17]:
```
meta_data.iloc[17863]['abstract']
```

Out[17]:
```
'Respiratory syncytial virus (RSV) is one of the main pathogens associated with lower respiratory tract infections in infants and young children worldwide. Exosomes secreted by antigen presenting cells (APCs) can elicit immune responses by carrying major histocompatibility complex (MHC) class I molecules complexed with antigenic peptides and other co-stimulating factors. Therefore, we developed novel immunomagnetic nanographene particles to sequentially isolate, surface engineer, and release intact dendritic cell (DC) exosomes for use as a potential vaccine platform against RSV. The H-2D(b)-restricted, immunodominant peptides from RSV M(M187-195) and NS1(61-75) were introduced to MHC-I on DC-derived exosomes to express peptide/MHC-I (pMHC-I) complexes. A mouse model of RSV infection was used to define the immunogenicity of surface engineered exosomes for activating virus-specific immune responses. Ex vivo assays demonstrated that engineered exosomes carrying RSV-specific peptides can elicit interferon-gamma (IFN-γ) production by virus-specific CD8+ T cells isolated from RSV-infected C57BL/6 mice. In vivo assays demonstrated that subcutaneous administration of both M(187-195) and NS1(61-75) engineered exosomes to mice, with or without additional adjuvant, appeared safe and well tolerated, however, did not prime antigen-specific CD8+ T cell responses. Surface engineered exosomes are immunogenic and promising for further development as a vaccine platform.'
```

In [5]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
wordcloud = WordCloud().generate(meta_data.iloc[17863]['abstract'])
plt.imshow(wordcloud, interpolation='bilinear')
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x7fe49035f4f0>
```



In [ ]:
```

```