Health news Recommendation system using Multilevel topic modeling

By
PRAVEEN KUMAR(CSE/16028/184)
MD SAIF UDDIN(CSE/16018/175)

Bachelor Thesis submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfillment of the degree of

Bachelor of Technology

in

Computer Science and Engineering/Information Technology

May, 2019

Certificate

*This is to certify that the thesis entitled "**Health news recommendation system using multilevel topic modeling** " being submitted by Praveen kumar,Md. Saif uddin an undergraduate student, Reg. No 184,175 in the Department of Computer Science and Information Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, for the award of Bachelors of Technology in Computer Science and Information Engineering /Information Technology is an original research work carried by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of Indian Institute of Information Technology Kalyani and in my opinion, has reached the standards needed for submission. The work,*

techniques and the results presented have not been submitted to any other University or Institute for the award of any other degree or diploma.

Dr. Sanjay Chatterji

Ph.D. [IIT Kharagpur]

Dept : Computer Science and Engineering

Indian Institute of Information Technology, Kalyani

Declaration

*I hereby declare that the work being presented in this thesis entitled, "**Health news recommendation system using multilevel topic modeling**", submitted to Indian Institute of Information Technology Kalyani in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering during the period from July, 2018 to May, 2019 under the supervision of **Dr.Sanjay Chatterji** , Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.*

PRAVEEN KUMAR (Roll No.-CSE/16028/184)

SAIF UDDIN(Roll No.-CSE/16028/175)

Name of the Department: Computer Science and Engineering

*Institute Name: **Indian Institute of Information Technology, Kalyani, W.B***

Acknowledgments

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, ***Dr. Sanjay Chatterji*** *for his invaluable advice, guidance and his enormous patience throughout the development of the research project.*

In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement.

PRAVEEN KUMAR (Roll No.-CSE/16028/184)

SAIF UDDIN(Roll No.-CSE/16018/175)

Name of the Department: ***Computer Science and Engineering***

Institute Name: Indian Institute of Information Technology, Kalyani, W.B

Place: Kalyani

Date:07/05/2019

Introduction

MOTIVE OF PROJECT:

Main motive to start this project is to recommend people about their daily health issue due to their mistake they do in their daily life for example 1:most of the people taken medicine without watching expiry date of medicine and due to that they have to face side effect of medicine . 2: Even though we are living in 21st century now also some people used to take medicine by themself without taken permission of doctor so that also leads to bad effect of health on people .

So here we tried to make people aware about these mistake what they do in their daily life using recommendation system on news data.

Features of dataset:

We have taken news data that includes data related to health ,sports,attacks,articles etc.. we tried to make dataset much general our results might be differ for different dataset .

Dataset link

Why we have taken news data

In daily newspaper we have seen that some news related to people have died due to wrong medicine they have taken without permission of doctor and also some people used to take medicine without watching expiry date of medicine.

Quick result for searching health data

We have developed a technique to find health related information from news paper or article without giving much time to read news paper or article .

Recommendation in health data(Unsupervised learning)

Recommendation system are utilized in variety of areas including news,research articles,social tag ,movie,etc.. Here we will use recommendation to aware the people about their daily mistake that affects their daily life on the basis of daily newspaper news (news data) or article news .

ex-In news paper one news (or Article) or research came that peoples are habituated to take medicine without permission of doctor that leads to bad effect on people health.

Ex- Sometimes before we have seen that one research had published that people are taking overdose of paracetamol that may increase their body immunity that's why same dose of paracetamol won't affect on disease since next time . **more about recommendation**

Topic modeling

Topic modeling is an unsupervised machine learning method that helps us discover hidden semantic structures in a paper, that allows us to learn topic representations of papers in a corpus. The model can be applied to any kinds of labels on documents, such as tags on posts on the website.

Abstract : Topic Modeling provides a convenient way to analyze big unclassified text. A topic contains a cluster of words that frequently occurs together. A topic modeling can connect words with similar meanings and distinguish between uses of words with multiple meanings. Topic modeling provides us methods to organize understand and summarize large collection of textual data like newspaper , research article etc.... .

·The methods of Topic modelling

A.Latent Dirichlet Allocation

The reason of appearance of Latent Dirichlet Allocation (LDA) model is to improve the way of mixture models that capture the exchangeability of both words and documents from the old way by PLSA and LSA. This was happened in 1990, so the classic representation theorem lays down that any collection of exchangeable random variables has a representation as a mixture distribution—in general an infinite mixture. There are huge

numbers of electronic document collections such as the web, scientifically interesting blogs, news articles and literature in the recent past has posed several new challenges to researchers in the data mining community. Especially there is a growing need of automatic techniques to visualize, analyze and summarize these document collections. In the recent past, latent topic modeling has become very popular as a completely unsupervised technique for topic discovery in large document collections. This model, such as LDA .

Latent Dirichlet Allocation (LDA) is an Algorithm for text mining that is based on statistical (Bayesian) topic models and it is very widely used. LDA is a generative model that tries to mimic what the writing process is. So it tries to generate a document on the given topic. It can also be applied to other types of data. There are tens of LDA based models including: temporal text mining, author- topic analysis, supervised topic models, latent Dirichlet co-clustering and LDA based bio_informatics .

Here we are using temporal text mining and latent Dirichlet co -clustering for getting Topic among all data and for better visualization co-clustering is used .

- **How do we use LDA**

The Process

- We pick the number of topics ahead of time even if we're not sure what the topics are.

- Each document is represented as a distribution over topics.

* Each topic is represented as a distribution over words.

The research paper text data is just a bunch of unlabeled texts and can be found here.

Different platform where we can work efficiently
If you are using jupyter notebook on i3 or i5 intel processor it might have take more time to run and also you might face installation problem of different packages regarding their version for example - scipy is available for specific version of pandas ,
So we suggest you to move towards Google colaboratory that is more likely as jupyter notebook and also GPU is available there so you can save your time in running these all stuff there and still you can do these thing in your jupyter notebook.

Before you start in google colaboratory run this below program to upload data set from your local system.
‘

```
from google.colab import files
uploaded = files.upload()
```

‘

Preprocessing

1. Tokenization

   Tokenization describes splitting paragraphs into sentences, or sentences into individual words.

2. Lemmatization

   It's used to convert different form of word form in their root form .

   Ex- cat's,cats -->cat

   Am ,are,is-->be

3. Stemming

Stem (root) is the part of the word to which you add inflectional (changing/deriving) affixes such as (-ed,-ize, -s,-de,mis). So stemming a word or sentence may result in words that are not actual words. Stems are created by removing the suffixes or prefixes used with a word

After preprocessing results are:

```
#@title Default title text
processed_docs = documents['headline_text'].map(preprocess)
processed_docs[:10]
```

```
0              [decide, community, broadcast, licence]
1                          [witness, aware, defamation]
2          [call, infrastructure, protection, summit]
3                           [staff, aust, strike, rise]
4                [strike, affect, cancern, travellers]
5               [ambitious, olsson, win, triple, jump]
6               [antic, delight, record, break, barca]
7    [aussie, qualifier, stosur, waste, memphis, ma...
8    [aust, address, security, council, smart, peop...
9                             [cancer, lock, timetable]
Name: headline_text, dtype: object
```

After preprocessing we form a dictionary in which we give every unique word a unique integer id .

```
[ ]  dictionary = gensim.corpora.Dictionary(processed_docs)
     count = 0
     for k, v in dictionary.iteritems():
         print(k, v)
         count += 1
         if count > 10:
             break
```

```
0 broadcast
1 community
2 decide
3 licence
4 aware
5 defamation
6 witness
7 call
8 infrastructure
9 protection
10 summit
```

After that finding unique word in dictionary:

```
bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
bow_corpus[4310]
```

```
[(118, 1), (501, 1), (980, 1), (4411, 1)]
```

So here id number 118 ,501,980,4411 having frequency 1 in entire corpus.

These unique integer are:

```
[12]  bow_doc_4310 = bow_corpus[4310]
      for i in range(len(bow_doc_4310)):
          print("Word {} (\"{}\") appears {} time.".format(bow_doc_4310[i][0],
                                                           dictionary[bow_doc_4310[i][0]],
      bow_doc_4310[i][1]))
```

```
Word 118 ("help") appears 1 time.
Word 501 ("rain") appears 1 time.
Word 980 ("bushfires") appears 1 time.
Word 4411 ("dampen") appears 1 time.
```

Till now we did all required process for running LDA on the data.So now we will run lda here.

```
[14] lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics=50, id2word=dictionary, passes=2, workers=2)
     lda_model.save('model50.gensim')
```

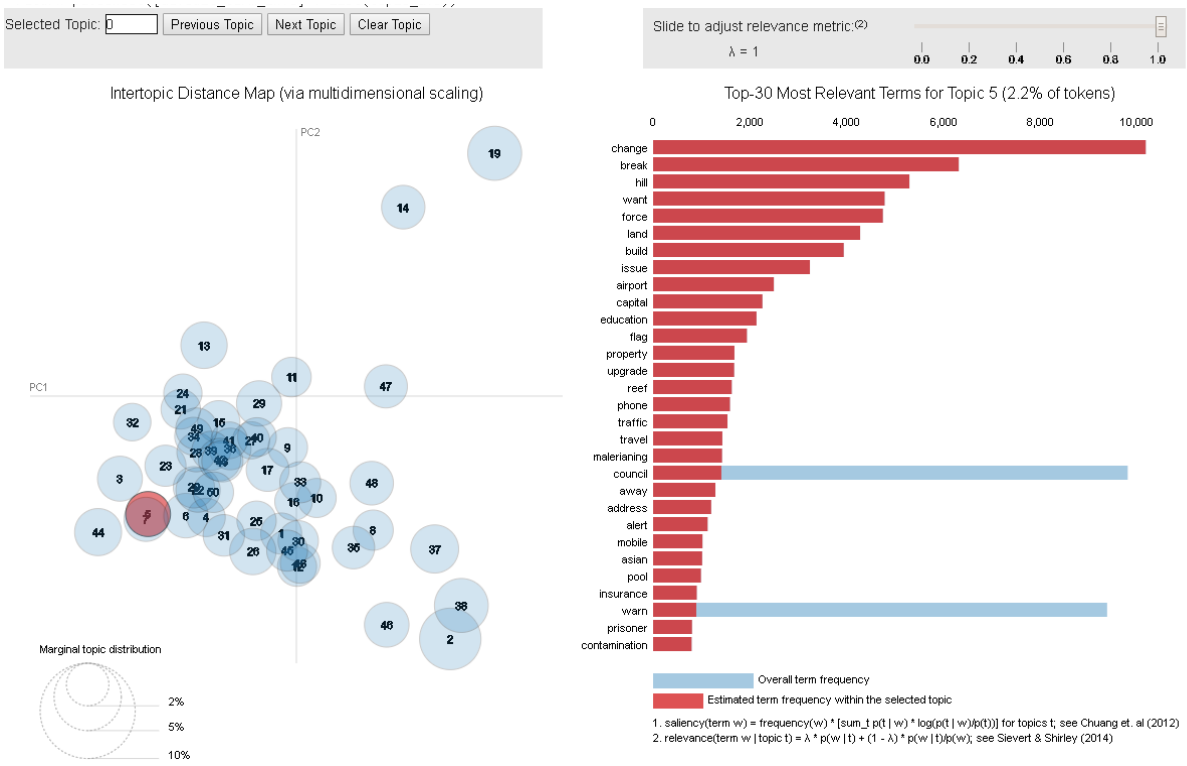
We run LDA on 50 topics and we saved the model into model50.gensim .
At topic 50 we got the most synchronised result towards health topic.

```
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))
```

```
Topic: 0
Words: 0.074*"national" + 0.058*"park" + 0.056*"meet" + 0.030*"great" + 0.029*"care" + 0.025*"clear" + 0.025*"d
Topic: 1
Words: 0.273*"cancer" + 0.159*"stage" + 0.149*"malignant" + 0.145*"benign" + 0.050*"interview" + 0.018*"officer
Topic: 2
Words: 0.099*"world" + 0.063*"labor" + 0.059*"concern" + 0.041*"industry" + 0.038*"residents" + 0.036*"trade" +
Topic: 3
Words: 0.055*"trump" + 0.043*"study" + 0.041*"david" + 0.032*"cancers" + 0.030*"mount" + 0.024*"victory" + 0.02
Topic: 4
Words: 0.093*"change" + 0.058*"break" + 0.048*"hill" + 0.044*"want" + 0.043*"force" + 0.039*"land" + 0.036*"bui
Topic: 5
Words: 0.065*"farm" + 0.062*"accuse" + 0.049*"review" + 0.042*"strike" + 0.042*"hobart" + 0.037*"appeal" + 0.03
Topic: 6
Words: 0.091*"report" + 0.060*"child" + 0.055*"budget" + 0.047*"victoria" + 0.046*"release" + 0.043*"public" +
Topic: 7
Words: 0.138*"symptom" + 0.132*"diesease" + 0.070*"brisbane" + 0.057*"leave" + 0.029*"bring" + 0.029*"grand" +
Topic: 8
Words: 0.043*"make" + 0.037*"rain" + 0.036*"michael" + 0.033*"dairy" + 0.032*"hop" + 0.031*"bushfire" + 0.028*"
Topic: 9
Words: 0.104*"house" + 0.063*"dead" + 0.046*"western" + 0.043*"northern" + 0.037*"battle" + 0.032*"boat" + 0.02
Topic: 10
```

In first level topic modeling we got the expected topic of entire documents and in 2nd level we try to get the topic related to health only and in next level we divide this topic in some special group in that this topic belongs to.
2nd level LDA we will run on these above topic.
Here we are plotting graph that will give better understanding of topics in whole documents.

So let’s see here when our data is synchronised.



2nd level topic modeling

For 2nd level topic modeling we will use topic of 1st level topic modeling (where we run first time LDA).

Here also we do everything same like preprocessing ,and dictionary making then we run LDA on that topic but here we are having less number of data.In first level of topic modeling we got the expected topic of entire documents and in second level topic modeling we got the topic related to health only and in further next level we assigned the group which this topic belong to.

Result of 2nd level topic modeling

Here we got the topic related to health only

```
[72] lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics=10, id2word=dictionary, passes=2, workers=2)
     lda_model.save('model10.gensim')

for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))

Topic: 0
Words: 0.046*"doctor" + 0.046*"campaign" + 0.046*"precautions" + 0.045*"cost" + 0.045*"effect" + 0.005*"attempt" + 0.005*"justice" + 0.005*"medicine" + 0.004*"
Topic: 1
Words: 0.046*"doctor" + 0.046*"cost" + 0.045*"campaign" + 0.045*"effect" + 0.045*"precautions" + 0.025*"medicine" + 0.024*"government" + 0.024*"cure" + 0.024*"
Topic: 2
Words: 0.117*"cure" + 0.117*"government" + 0.116*"concern" + 0.116*"family" + 0.116*"support" + 0.116*"medicine" + 0.116*"symptom" + 0.004*"campaign" + 0.004*"
Topic: 3
Words: 0.012*"precautions" + 0.012*"doctor" + 0.012*"cost" + 0.012*"campaign" + 0.012*"effect" + 0.009*"cure" + 0.009*"medicine" + 0.009*"government" + 0.009*"
Topic: 4
Words: 0.033*"precautions" + 0.033*"doctor" + 0.033*"cost" + 0.033*"campaign" + 0.033*"effect" + 0.007*"symptom" + 0.005*"family" + 0.005*"concern" + 0.005*"me
Topic: 5
Words: 0.048*"effect" + 0.048*"campaign" + 0.047*"doctor" + 0.047*"cost" + 0.047*"precautions" + 0.023*"symptom" + 0.023*"medicine" + 0.022*"family" + 0.022*"c
Topic: 6
Words: 0.045*"cost" + 0.044*"doctor" + 0.044*"effect" + 0.044*"precautions" + 0.044*"campaign" + 0.016*"cure" + 0.016*"medicine" + 0.016*"government" + 0.016*"
Topic: 7
Words: 0.039*"campaign" + 0.037*"precautions" + 0.037*"effect" + 0.036*"doctor" + 0.036*"cost" + 0.009*"support" + 0.007*"family" + 0.007*"concern" + 0.007*"sy
Topic: 8
Words: 0.038*"precautions" + 0.038*"effect" + 0.038*"doctor" + 0.038*"cost" + 0.037*"campaign" + 0.007*"family" + 0.007*"concern" + 0.005*"cure" + 0.005*"medic
Topic: 9
Words: 0.043*"cost" + 0.042*"precautions" + 0.042*"effect" + 0.042*"campaign" + 0.042*"doctor" + 0.004*"book" + 0.004*"michael" + 0.004*"economy" + 0.004*"trac
```

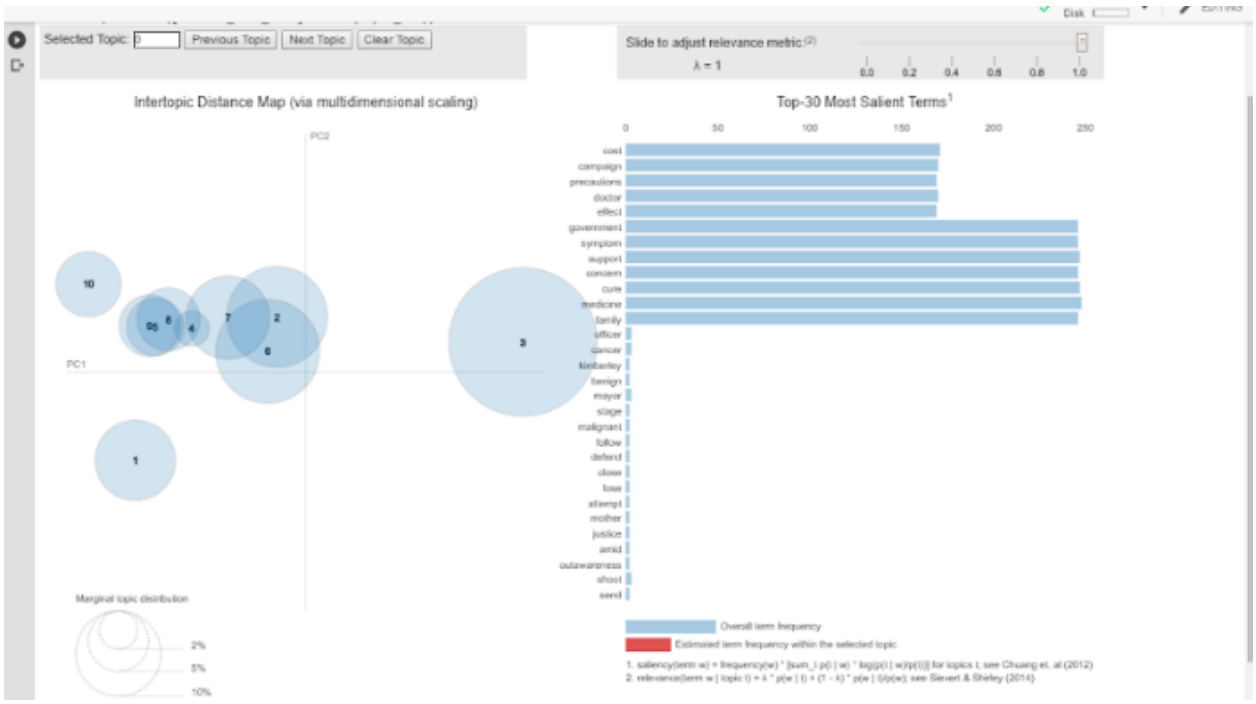
Ten topics related to health.

```
Topic: 0
Words: 0.046*"doctor" + 0.046*"campaign" + 0.046*"precautions" + 0.045*"cost" + 0.045*"effect" + 0.005*"attempt" +
0.005*"justice" + 0.005*"medicine" + 0.004*"family" + 0.004*"symptom"
Topic: 1
Words: 0.046*"doctor" + 0.046*"cost" + 0.045*"campaign" + 0.045*"effect" + 0.045*"precautions" + 0.025*"medicine" +
0.024*"government" + 0.024*"cure" + 0.024*"support" + 0.024*"concern"
Topic: 2
Words: 0.117*"cure" + 0.117*"government" + 0.116*"concern" + 0.116*"family" + 0.116*"support" + 0.116*"medicine" +
0.116*"symptom" + 0.004*"campaign" + 0.004*"effect" + 0.004*"precautions"
Topic: 3
Words: 0.012*"precautions" + 0.012*"doctor" + 0.012*"cost" + 0.012*"campaign" + 0.012*"effect" + 0.009*"cure" + 0.009*"medicine"
+ 0.009*"government" + 0.009*"support" + 0.009*"family"
Topic: 4
Words: 0.033*"precautions" + 0.033*"doctor" + 0.033*"cost" + 0.033*"campaign" + 0.033*"effect" + 0.007*"symptom" + 0.005*"family"
+ 0.005*"concern" + 0.005*"medicine" + 0.005*"government"
Topic: 5
Words: 0.048*"effect" + 0.048*"campaign" + 0.047*"doctor" + 0.047*"cost" + 0.047*"precautions" + 0.023*"symptom" +
0.023*"medicine" + 0.022*"family" + 0.022*"concern" + 0.022*"support"
Topic: 6
Words: 0.045*"cost" + 0.044*"doctor" + 0.044*"effect" + 0.044*"precautions" + 0.044*"campaign" + 0.016*"cure" + 0.016*"medicine"
+ 0.016*"government" + 0.016*"support" + 0.014*"symptom"
Topic: 7
Words: 0.039*"campaign" + 0.037*"precautions" + 0.037*"effect" + 0.036*"doctor" + 0.036*"cost" + 0.009*"support" + 0.007*"family"
+ 0.007*"concern" + 0.007*"symptom" + 0.006*"government"
Topic: 8
Words: 0.038*"precautions" + 0.038*"effect" + 0.038*"doctor" + 0.038*"cost" + 0.037*"campaign" + 0.007*"family" + 0.007*"concern"
+ 0.005*"cure" + 0.005*"medicine" + 0.005*"symptom"
```

Here topic 0 belongs to ‘Awareness’.

Topic 1,2,3,4,5,6,7,8 belongs to both ‘Government support’ and ‘awareness’.

Getting synchronised result at topic 10:



3rd level topic modeling

We run lda on result of 2nd level topic modeling and we make some group that this topic belongs . for example(Benign belongs to cancer group,society support belongs to environmental support etc....)

Here is some group we have created.

1.Awareness

2.Environmental support(social support)

3.Government support

4.Disease

5.

Results :

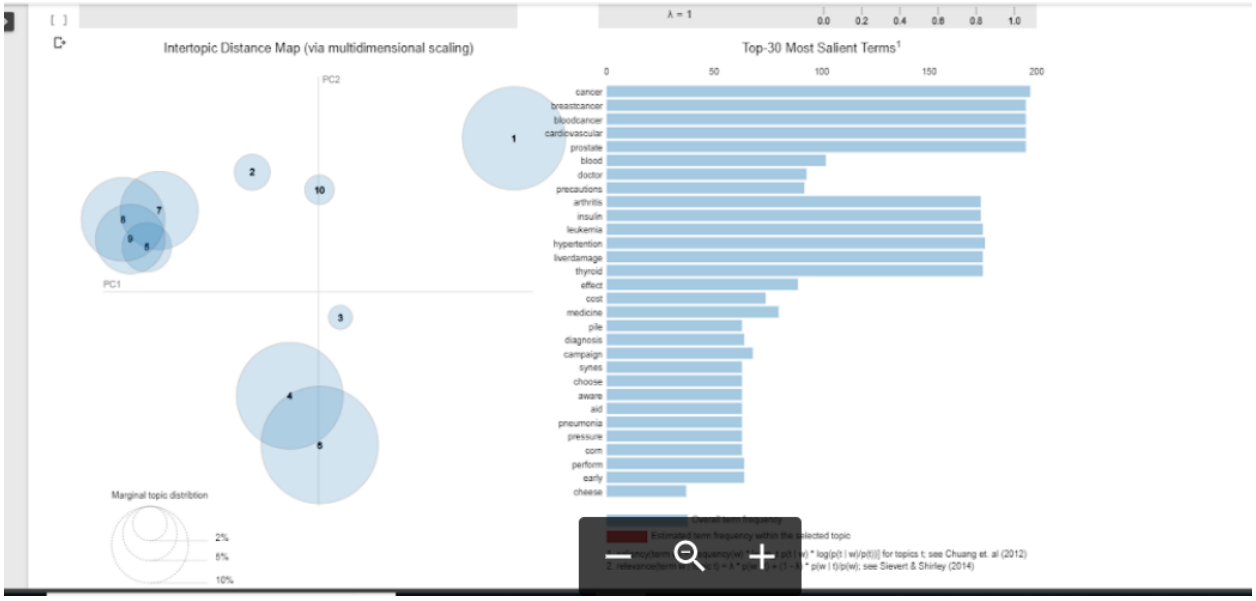
```
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))
```

```
Topic: 0
Words: 0.175*"breastcancer" + 0.175*"bloodcancer" + 0.175*"cancer" + 0.175*"cardiovascular" + 0.175*"prostate" + 0.002*"members" + 0.002*"custody" + 0.002*"coach" + 0.002*"rape" + 0.002*"leak"
Topic: 1
Words: 0.030*"arthritis" + 0.030*"leukemia" + 0.030*"doctor" + 0.029*"thyroid" + 0.029*"liverdamage" + 0.029*"hypertention" + 0.029*"precautions" + 0.029*"insulin" + 0.027*"bloodcancer" + 0.027*"breas
Topic: 2
Words: 0.036*"cancer" + 0.035*"breastcancer" + 0.035*"cardiovascular" + 0.035*"bloodcancer" + 0.035*"prostate" + 0.031*"blood" + 0.023*"arthritis" + 0.023*"insulin" + 0.022*"hypertention" + 0.022*"leu
Topic: 3
Words: 0.039*"liverdamage" + 0.039*"hypertention" + 0.038*"insulin" + 0.038*"arthritis" + 0.037*"thyroid" + 0.037*"leukemia" + 0.033*"medicine" + 0.033*"support" + 0.032*"symptom" + 0.032*"government"
Topic: 4
Words: 0.045*"cost" + 0.044*"effect" + 0.044*"doctor" + 0.043*"thyroid" + 0.043*"arthritis" + 0.043*"leukemia" + 0.043*"hypertention" + 0.043*"insulin" + 0.043*"liverdamage" + 0.040*"campaign"
Topic: 5
Words: 0.060*"blood" + 0.033*"thyroid" + 0.033*"medicine" + 0.033*"leukemia" + 0.033*"hypertention" + 0.032*"arthritis" + 0.032*"insulin" + 0.032*"liverdamage" + 0.031*"aid" + 0.031*"corn"
Topic: 6
Words: 0.056*"insulin" + 0.055*"liverdamage" + 0.055*"hypertention" + 0.055*"leukemia" + 0.054*"thyroid" + 0.053*"arthritis" + 0.048*"precautions" + 0.047*"doctor" + 0.039*"cost" + 0.036*"effect"
Topic: 7
Words: 0.059*"precautions" + 0.058*"effect" + 0.057*"thyroid" + 0.057*"arthritis" + 0.057*"hypertention" + 0.056*"liverdamage" + 0.056*"leukemia" + 0.056*"insulin" + 0.052*"doctor" + 0.048*"campaign"
Topic: 8
Words: 0.057*"arthritis" + 0.056*"thyroid" + 0.056*"leukemia" + 0.055*"liverdamage" + 0.055*"hypertention" + 0.055*"insulin" + 0.046*"doctor" + 0.042*"effect" + 0.037*"precautions" + 0.030*"cost"
Topic: 9
Words: 0.049*"cancer" + 0.039*"bloodcancer" + 0.039*"prostate" + 0.038*"breastcancer" + 0.038*"cardiovascular" + 0.016*"leukemia" + 0.016*"arthritis" + 0.016*"thyroid" + 0.015*"insulin" + 0.015*"liver
```

```
Topic: 0
Words: 0.175*"breastcancer" + 0.175*"bloodcancer" + 0.175*"cancer" +
0.175*"cardiovascular" + 0.175*"prostate" + 0.002*"members" + 0.002*"custody"
+ 0.002*"coach" + 0.002*"rape" + 0.002*"leak"
Topic: 1
Words: 0.030*"arthritis" + 0.030*"leukemia" + 0.030*"doctor" +
0.029*"thyroid" + 0.029*"liverdamage" + 0.029*"hypertention" +
0.029*"precautions" + 0.029*"insulin" + 0.027*"bloodcancer" +
0.027*"breastcancer"
Topic: 2
Words: 0.036*"cancer" + 0.035*"breastcancer" + 0.035*"cardiovascular" +
0.035*"bloodcancer" + 0.035*"prostate" + 0.031*"blood" + 0.023*"arthritis" +
0.023*"insulin" + 0.022*"hypertention" + 0.022*"leukemia"
Topic: 3
Words: 0.039*"liverdamage" + 0.039*"hypertention" + 0.038*"insulin" +
0.038*"arthritis" + 0.037*"thyroid" + 0.037*"leukemia" + 0.033*"medicine" +
0.033*"support" + 0.032*"symptom" + 0.032*"government"
Topic: 4
Words: 0.045*"cost" + 0.044*"effect" + 0.044*"doctor" + 0.043*"thyroid" +
0.043*"arthritis" + 0.043*"leukemia" + 0.043*"hypertention" + 0.043*"insulin"
+ 0.043*"liverdamage" + 0.040*"campaign"
Topic: 5
```

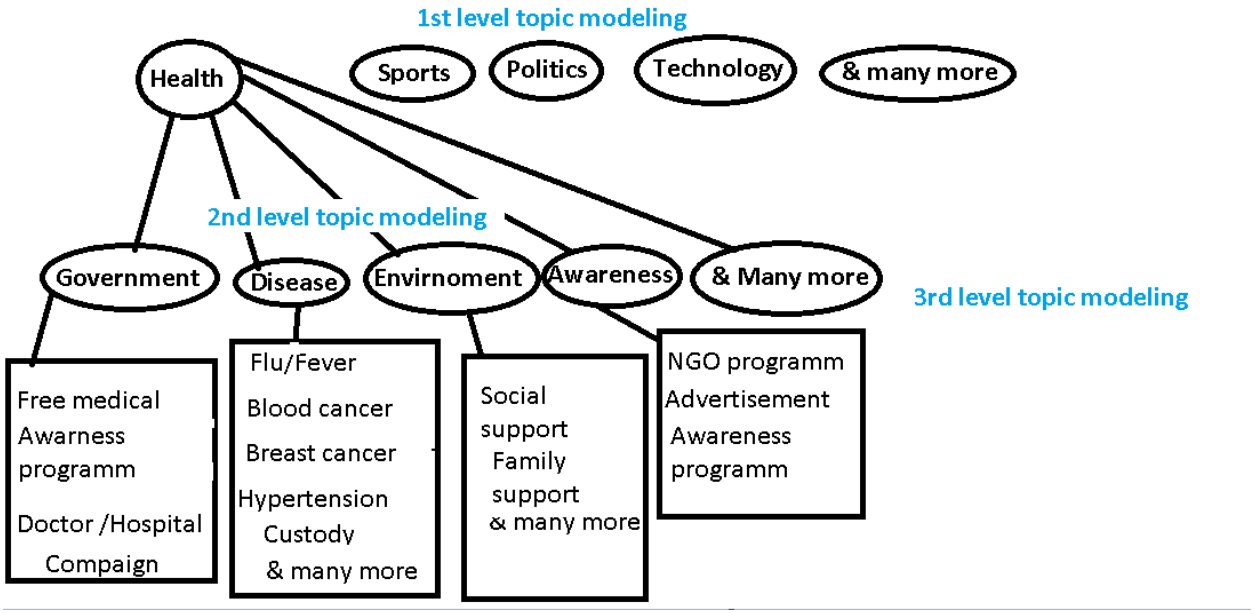
Topic 0,1,2 purely based on ‘Disease’ and rest of them are belongs to both ‘Disease’ and ‘Government support’.

Graphical representation of above topic:



RESULTS:

Generalization of multilevel topic modeling:



Here we categorized every topic in some specific domain
That will give clarity from which domain we have to recommend to the people.
In multilevel topic modelling we categorized all the topic in their specific topic domain.

Conclusion:

In 2nd level topic modeling we did for disease(domain) we can do same for Government ,environment ,awareness and many more .
Above i have plotted diagram of expected domain.
These above domain we got from 2nd level topic modeling and from there we have specific domain where we can apply further topic modeling to get interrelated topic that would be helpful in recommendation .

For ex- if we have to recommend people about awareness about any disease then we would use awareness domain .
If government is giving any type supports to any disease patient then we can use government support domain for make them aware to the people.