Topic Modeling Explanation

Topic Modeling using LDA

Topic Modeling is a type of statistical modeling to discover abstract topics that are in a set of documents. It uses statistical language models to uncover hidden structure in a collection of texts. The main goal is to find out what are the topics that are in a set of documents.

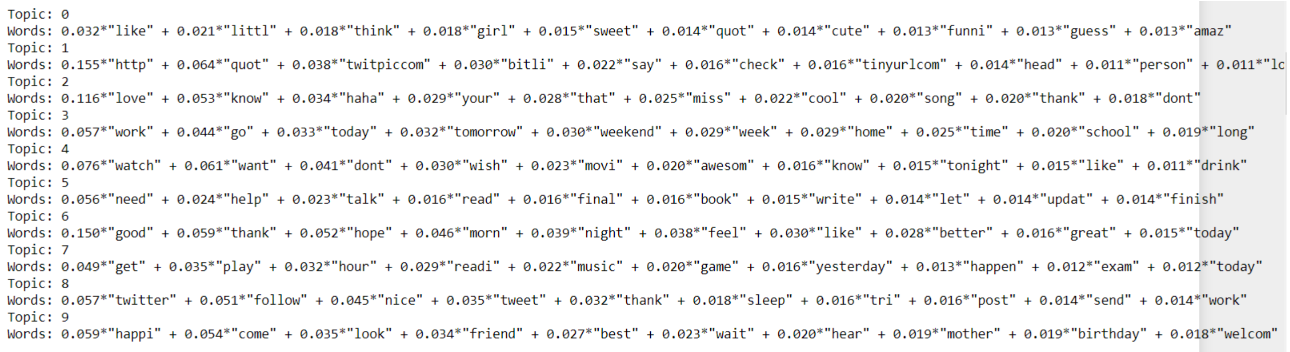
In my project, “Twitter Tweets Topic Modeling”, I used LDA to do topic modeling and used Gensim package.

Latent Dirichlet Allocation (LDA) is an example of topic modeling. It generates topics based on frequency of words from a set of documents.

Gensim is a Python library for topic modeling using natural language processing.

Behind LDA:

1. LDA randomly assign topics depending on how many topics that was chosen. In my case, I picked 10 topics with 10 words each
2. Looks for frequency of words associated to that topic. It looks at the whole document and look for the frequency of the word in the topic
3. Once it gets the number of words in a topic, it reassigns topics depending on word counts.
4. Removes current assignment (word) and decrement the word count.
5. Calculates (multiply a and b to get the probability) :
    a. How much the document likes each topic based on word
    b. How much topic likes the word
6. Repeats the process for a large number of times



```
Topic: 0
Words: 0.032*"like" + 0.021*"littl" + 0.018*"think" + 0.018*"girl" + 0.015*"sweet" + 0.014*"quot" + 0.014*"cute" + 0.013*"funni" + 0.013*"guess" + 0.013*"amaz"
Topic: 1
Words: 0.155*"http" + 0.064*"quot" + 0.038*"twitpiccom" + 0.030*"bitli" + 0.022*"say" + 0.016*"check" + 0.016*"tinyurlcom" + 0.014*"head" + 0.011*"person" + 0.011*"lc
Topic: 2
Words: 0.116*"love" + 0.053*"know" + 0.034*"haha" + 0.029*"your" + 0.028*"that" + 0.025*"miss" + 0.022*"cool" + 0.020*"song" + 0.020*"thank" + 0.018*"dont"
Topic: 3
Words: 0.057*"work" + 0.044*"go" + 0.033*"today" + 0.032*"tomorrow" + 0.030*"weekend" + 0.029*"week" + 0.029*"home" + 0.025*"time" + 0.020*"school" + 0.019*"long"
Topic: 4
Words: 0.076*"watch" + 0.061*"want" + 0.041*"dont" + 0.030*"wish" + 0.023*"movi" + 0.020*"awesom" + 0.016*"know" + 0.015*"tonight" + 0.015*"like" + 0.011*"drink"
Topic: 5
Words: 0.056*"need" + 0.024*"help" + 0.023*"talk" + 0.016*"read" + 0.016*"final" + 0.016*"book" + 0.015*"write" + 0.014*"let" + 0.014*"updat" + 0.014*"finish"
Topic: 6
Words: 0.150*"good" + 0.059*"thank" + 0.052*"hope" + 0.046*"morn" + 0.039*"night" + 0.038*"feel" + 0.030*"like" + 0.028*"better" + 0.016*"great" + 0.015*"today"
Topic: 7
Words: 0.049*"get" + 0.035*"play" + 0.032*"hour" + 0.029*"readi" + 0.022*"music" + 0.020*"game" + 0.016*"yesterday" + 0.013*"happen" + 0.012*"exam" + 0.012*"today"
Topic: 8
Words: 0.057*"twitter" + 0.051*"follow" + 0.045*"nice" + 0.035*"tweet" + 0.032*"thank" + 0.018*"sleep" + 0.016*"tri" + 0.016*"post" + 0.014*"send" + 0.014*"work"
Topic: 9
Words: 0.059*"happi" + 0.054*"come" + 0.035*"look" + 0.034*"friend" + 0.027*"best" + 0.023*"wait" + 0.020*"hear" + 0.019*"mother" + 0.019*"birthday" + 0.018*"welcom"
```

Interpretation of results:

Topic 0 has largest weight of 0.032. This is the probability of the word “like”. Looking at 10 topics that has been generated we can say something about what topic is about. For topic 0, I can say that it has something to do with characteristics of a girl given that it has words “little”, “sweet”, “cute”, “funny”, “amaze”. The words “like” and “quot” does not make much sense in topic 0. I can say that we could clean the dataset a little more by removing these words. We can include these words when we remove stop words. Topic 1 has websites of emojis or gifs. We can say that these tweets contain emojis and gifs. Topic 2 is a little tricky, but I can say that it has something to do about love. Topic 3 has something to do about work, home and school. Topic 4 has something to do about watching a movie. Topic 5 is about reading a book. Topic 6 about greetings like “good morning”, or “good night”. Topic 7 is about activities like playing a game, reading, music and exam. Topic 8 has something to do about Twitter or updating Twitter. Topic 9 has something to do about birthday because most of the words in this topic can be said when it’s somebody’s birthday.

By looking at the words and the weights in each topic, we can say something about what the topic is about.

LDA Visualization through pyLDAvis

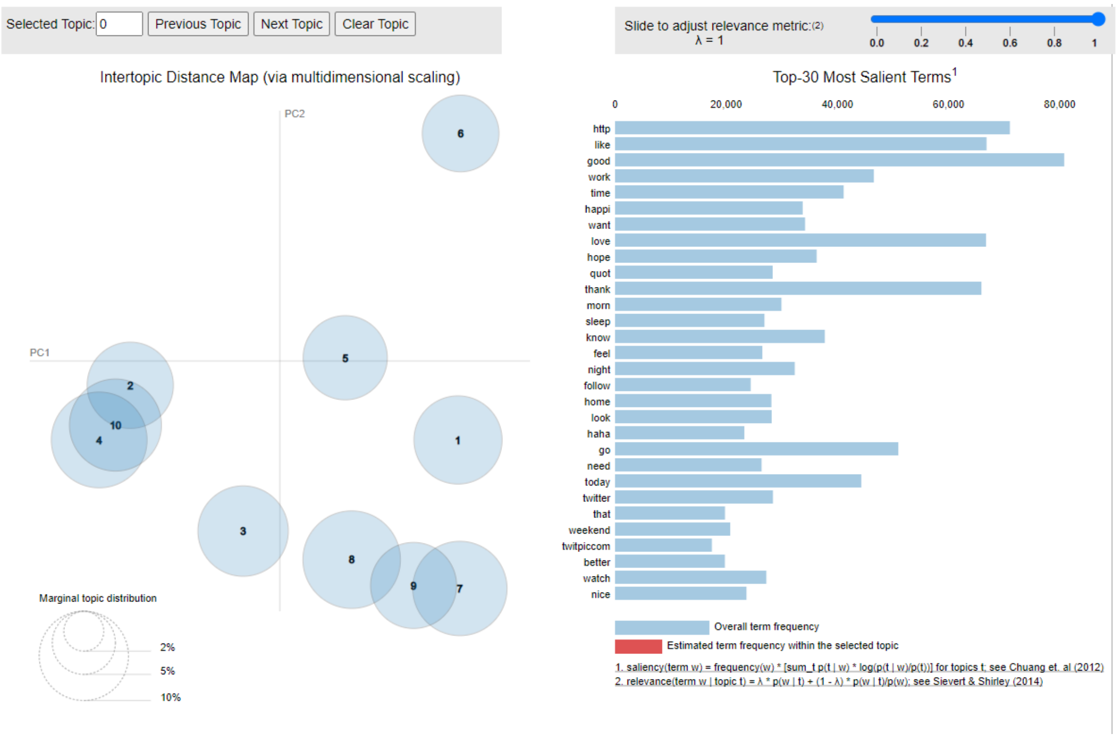
LDA visualization is helpful to interpret the topics in a topic model.

pyLDAvis is a Python library that helps users to visualize and interpret what a particular topic is about. It extracts information from a fitted LDA topic model to inform interactive web-based visualization.

I referred to this site to learn this: (https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21)

- Saliency – measure of how much the term tells you about the topic
  - Terms that mostly tells us about what’s going on relative to the topics
- Relevance – denoted by lambda
  - Weighted average of the probability of the word given the topic is normalized by the probability of the topics.
    - The goal of normalization is to **change the values of numeric columns** in the dataset to a common scale, without distorting differences in the ranges of values.
    - Sometimes normalization can make the accuracy better
  - **Has something to do on changing lamda to achieve the accuracy of the weight or proportion to be able to see which terms are really relevant and rate them**

    - Denotes the degree to which each term appears in a particular topic to the exclusion of others
    - It is the weight assigned to the probability of a term in a topic relative to its *lift* (ratio of a term's probability within a topic to its margin probability across the corpus)
    - λ = 1, the terms are ranked by their probabilities within the topic (the 'regular' method – overall term frequency)
        - "overall term frequency" gets higher
    - λ = 0, the terms are ranked only by their lift (estimated term frequency within the selected topic)
        - only the "estimated term frequency within the selected topic" is computed
    - As relevance metric gets higher, overall term frequency get higher
    - https://www.objectorientedsubject.net/2018/08/experiments-on-topic-modeling-pyldavis/
- Size of the bubble – measure the importance of the topics relative to the data
    - Bubbles that are clustered – doesn't mean they are similar topics. Our graph is a multidimensional scaling which means that in other dimension, they are not necessary clustered.
        - It simply means that this particular topic is near the other topics
- "Top 30 Most Relevant Terms"
    - Relevant terms to a particular topic
    - The words on LDA that has weights they are the top terms or top words



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Interpretation of results:

Looking at the left graph, some bubbles are overlapping. Topics 2, 4 and 10 are clustered together and topics 7, 8, and 9 are clustered together. Clustered bubbles do not mean they have same topic. They are only clustered because we are using multidimensional scaling. Each dimension is like a vector. They are not necessarily overlapping in another dimension.

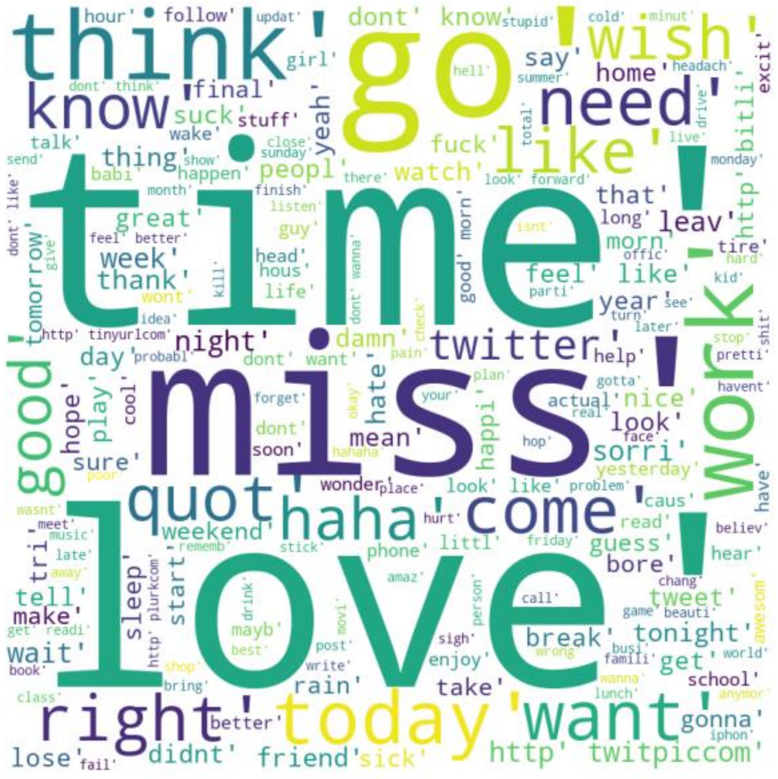
Looks like the bubbles are well distributed even though some overlapped. In terms of size, all of them are pretty close to each other which means that all topics are pretty important relative to the data.

The bar graph on the right side is the top 30 salient terms. When you hover over to the bubble, the bar graph on the right will give you the top 30 salient terms. These are the words that are relative to the topic as well. It can help us determine more exact conclusion on what the topic is all about.

Word Cloud

Word cloud is a technique for visualizing text data. The size of each word indicates the word's frequency and importance in the dataset. The bigger the word in the word cloud, the higher number of frequency and importance.

Matplotlib and **wordcloud** library are the packages I used to make a word cloud.



Interpretation of results:

The words "time" "miss", and "love" are the most frequent and important words in the dataset. This means that the tweets contain these three words a lot. The reason with this is because of wordcloud library. The wordcloud library is the one who looks at the tweets and keep track of the frequent and important words in the dataset.