Topic Modeling using LDA

Submitted to: Minapharm

Author: Eslam Abdelghany

Eslam322_1@hotmail.com

Problem statement

A medical-related dataset, containing 5k articles. It is required to map each document to the top 3 topics it belongs to and show the probability score for each topic. I solved this problem using latent Dirichlet allocation (LDA) *Genism* implementation.

Tools used:

- Jupyter notebook
- Python
- NLTK
- Spacy
- Genism
- Pandas
- Matplotlib
- Numpy

The followed Approach

- **Data Preprocessing:** this step is used to prepare the raw text for feature extraction. The process included:
  1- Removing Punctuation and lowering case all words.
  2- Removing stop words that do not affect the overall meaning of a sentence such as he, she, etc...

3- Lemmatization to return words to their root meaning following English dictionary such as (ran, run),(are, be), and Stemming that does the same thing following an algorithm (focus, focu),(association, associ).

- **Feature extraction:** Machine learning algorithms do not deal with text it only understands numbers. So that is why the step of converting a document to numbers with meaningful features is crucial for training the algorithm. This can be done by :

  1- Converting the training set we have to a dictionary that tells how many times a word occurred in the dataset.

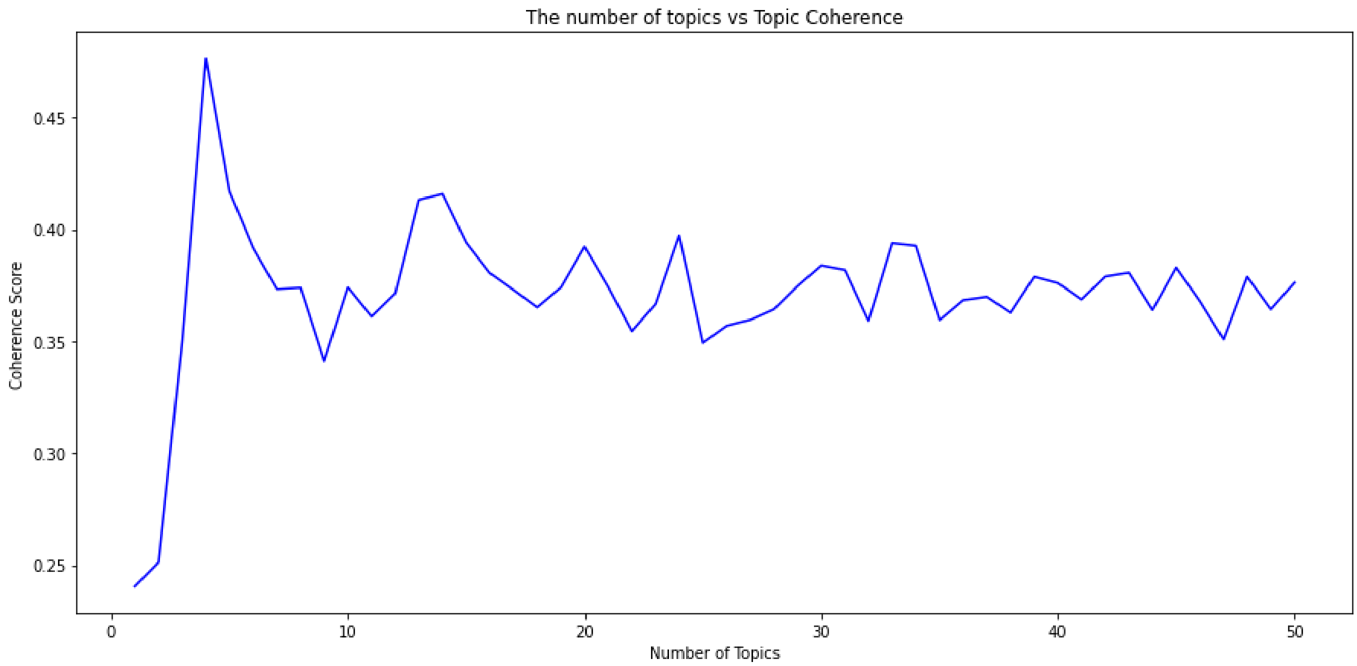
  2- Creating a bag of words BOW that tells how many times a word occurred within a document.

  3- TF-ID that instead of only counting the number of occurrences, it takes into account the frequency of occurrence of each word in a document. Hence, it tells how important a specific word is to a document.

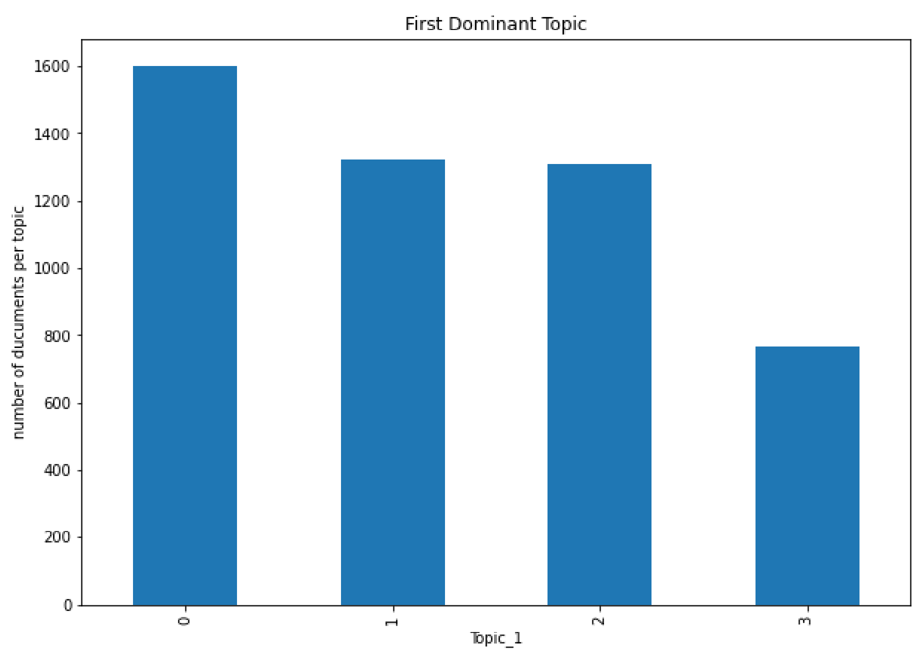
- Model training and choosing the optimal number of topics. I used topic coherence to evaluate the performance of the LDA following the same technique in this paper. Then, A CSV file is loaded including the assignment of each document to the top three topics it may belong to using a probability score to order them.
- Results Loading (Graphs, CSVs, and the final model)

Results and Findings:

After training 50 models using the number of topics as the only variable parameter and storing the coherence score of each model, I found that the number of topics that has the highest coherence score is 4. The results are illustrated in the following graph.



Among the three topics that a document may belong to, the following graph shows the first dominant topic among all documents.



Final Thoughts

I think if I included more feature extraction such as adding tri-grams and building the model without stemming, this may result in a better performance. The problem with stemming is that sometimes it results in some words that are not human-friendly or actual English vocabulary.