Topic Modeling in Natural Language Processing

Varun Chilukuri
Marian Qian

May 2019

1 Introduction

Natural Language Processing or NLP, is a branch of machine learning that attempts to read, decipher, and understand human languages. It sits at the intersection of computer science, artificial intelligence, and computational linguistics. NLP algorithms can be found in many applications, including Google Translate, Amazon Alexa, Google Home, and Microsoft Word. Topic modeling, a subset of the field of NLP involves processing text and identifying the topics present.

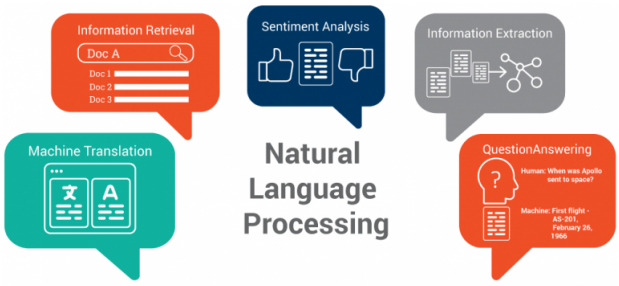

Figure 1: Uses for Natural Language Processing

2 Concepts

Latent Dirichlet Allocation (LDA) is the most popular technique used for topic modeling. Given a set of documents, it is used to determine which topics would most likely lead to the set of documents. LDA is a bag-of-words model that must use pre-processed data and assumes the topic distribution has a sparse Dirichlet prior. After LDA, a probability distribution of possible topics is assigned to the documents, while the probabilities of each word contained in the text for each topic assigned to the document is also produced. The probability distribution is

a Dirichlet distribution ($P(\theta|\overrightarrow{\alpha}) = \frac{\Gamma(\Sigma_i \alpha_i) f}{\Pi_i \Gamma(\alpha_i)} \Pi_i \theta_i^{\alpha_i - 1}$), but the math is beyond the scope of this lecture. It is also important to note that a word can appear in multiple topics with a different probability. The LDA model's plate notation is shown, where N is the number of words in a document, D is the number of documents, and K is the number of topics. The only observable variable is $W_{d,n}$, and topic $Z_{d,n}$ is assigned to the nth word in document d, with a probability θ_d from the topic distribution for document d drawn from a Dirichlet with parameter α. The same word $W_{d,n}$ is drawn from topic k with probability of β_k, where β_k is the Dirichlet word distributions per topic k using hyperparameter η.

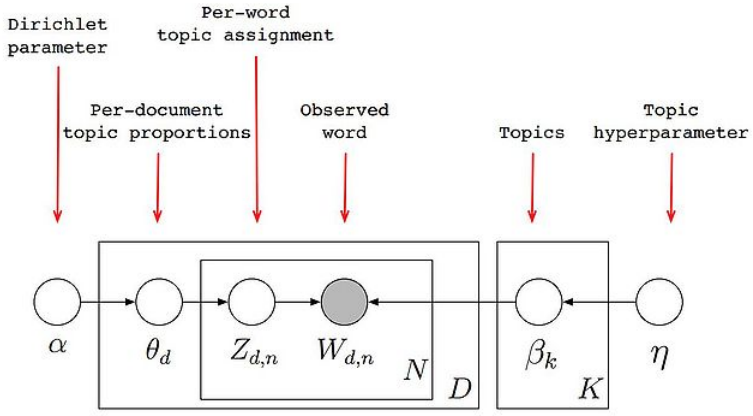

Figure 2: Graphical Representation of LDA model

2.1 Processing the Data

Processing the data includes many steps. All stop words in the original sentences must be removed. Stop words are common words in English that do not add meaning to a sentence. Examples include "a", "and", "but", "how", and "or". The sentence that remains must be tokenized, which means that it must be broken up into individual words. All punctuation is removed and all words are made lowercase. Next, the words must be lemmatized. All words in third person are changed to first person, while any verbs in past and future tenses are changed into present. Words must also be stemmed, which means that they are reduced to their root, or word stem. Here's an example:

> Original sentence: "The boy has cars that are different colors."
>
> After removing stop words: "Boy cars different colors."
>
> Tokenization: [boy, cars, different, colors]
>
> Lemmatization/Stemming: [boy, car, differ, color]

2.2 Matrix Factorization

A document-term matrix, which is essentially a frequency distribution must first be made. The following matrix represents the document-term matrix for n documents $D_1, D_2, D_3, ..., D_n$ and vocabulary size of m words $w_1, w_2, w_3, ..., w_m$. The value of $S_{i,j}$ in the matrix gives the frequency count of w_j in D_i.

$$\begin{array}{c} \\ \mathbf{S} = \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & w_2 & w_2 & ... & w_m \end{array} \\ \begin{bmatrix} 0 & 9 & 8 & ... & 7 \\ 3 & 5 & 6 & ... & 1 \\ 1 & 0 & 0 & ... & 2 \\ 1 & 0 & 0 & ... & 2 \end{bmatrix} \end{array} \begin{array}{c} \\ \\ D_1 \\ D_2 \\ D_3 \\ D_n \end{array}$$

Two other matrices must also be made. Matrix S is a document-term matrix, but document-topic and topic-term matrices are also necessary. If T represents each topic , k represents the number of topics, and v represents the vocabulary size, the following matrices can be created:

$$\begin{array}{c} \\ \mathbf{F} = \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{T}_1 & \mathrm{T}_2 & \mathrm{T}_2 & ... & \mathrm{T}_k \end{array} \\ \begin{bmatrix} 3 & 5 & 2 & ... & 9 \\ 4 & 8 & 6 & ... & 5 \\ 2 & 5 & 1 & ... & 7 \\ 0 & 2 & 8 & ... & 4 \end{bmatrix} \end{array} \begin{array}{c} \\ \\ D_1 \\ D_2 \\ D_3 \\ D_n \end{array}$$

$$\begin{array}{c} \\ \mathbf{G} = \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{w}_1 & \mathrm{w}_2 & \mathrm{w}_2 & ... & \mathrm{w}_v \end{array} \\ \begin{bmatrix} 5 & 8 & 7 & ... & 0 \\ 2 & 9 & 2 & ... & 6 \\ 7 & 4 & 1 & ... & 4 \\ 3 & 1 & 9 & ... & 8 \end{bmatrix} \end{array} \begin{array}{c} \\ \\ \mathrm{T}_1 \\ T_2 \\ T_3 \\ T_k \end{array}$$

F represents the document-topic matrix, topic distribution for the document, and G represents the topic-term matrix, the word distribution for the topic.

2.3 Conditional Probability

Given a set of documents, the probability distribution in LDA can be used to determine the probability of a word appearing in a given topic, P(word | topic). Finding the topic the document is about requires the probabilities of topics, given the document containing specific words, P(topic | word), and Bayes' rule will calculate this probability using P(word | topic) from Gibbs sampling. Probability of the document being about a topic given word

$$\overbrace{P(B|A)}^{\mathrm{P(topic \mid word)}} = \overbrace{P(A|B)}^{\mathrm{P(word \mid topic)}} * P(B)/P(A)$$

2.4 Collapsed Gibbs Sampling

Gibbs sampling is Markov chain Monte Carlo statistical inference algorithm commonly used to determine the probability of a word being assigned to a given topic. The algorithm finds P(word | topic) by learning the model from the given data and is used in LDA to calculate the probabilities. A simplified version of the algorithm is listed below:

Initialize and assign random topics to each word in set of documents
for n iterations **do**
  for D documents **do**
    for W words [1] **do**
      for T topic **do**
        p = probability of topic T assigned to word W
        p = num total words in topic T * P(word W | topic T)[2]
        W word's new assigned topic = topic with maximum value of p

Collapsed Gibbs sampling changes the topic distribution of the document by changing the respective word distributions for each topic. Because the algorithm generates data based off of random initialization, it is generally heavy computationally and requires a long time to run on large data sets. Current research about different variations of Gibbs sampling aim to increase efficiency and reduce number of calculations needed for the model to come to a conclusion regarding the topic distribution for a set of given documents.

3 Limitations of Topic-Modeling

Topic modeling produces the best results when applied to longer documents and those that have a consistent structure. It would not be as accurate with shorter documents, such as image captions or tweets. Like Naive Bayes, LDA also assumes that there is no correlation between different topics, which we know to be false. A document containing the topic "dog" is also more likely to also contain the topic "animal". This tendency is not effectively modeled by LDA.

4 Further Explorations

Topic Modeling with LDA is not only limited to texts. It can similarly be used with images as well. For further research, these libraries and resources might serve to be useful:

- Apache OpenNLP
- Natural Language Toolkit (NLTK)

[1] Assumes all other topic assignments for rest of words are correct
[2] From previous topic distribution

- Stanford NLP
- MALLET

5 Acknowledgements

The information and images used in this lecture came from multiple sources. All credit goes to these authors:

- Zholin Qiu, author of *Collapsed Gibbs Sampling for Latent Dirichlet Allocation on Spark*
- Prithu Banerjee of the University of British Columbia
- Chris Bail of Duke University
- Milena Yankova of OntoText
- Matt Kiser of Algorithmia