High user Experience by Providing Relevant News Articles using Topic Modelling

Santhosh Thiyagarajan[#1]
Master of Technology, Software Systems
Brila Institute of Technology, Pilani(Raj.), India.

***Abstract*—*The digital world where the data grows at an exponential rate where most of them are unstructured in nature. The major task is to categorize them for valuable data extraction. One of the best methods to structure the data is to put them under a topic. The advancement in the computer field gives us various ways to categorize the data corpus such as TF-IDF, MALLET, LDA and so on. Once the model is designed with the appropriate number of topics, then it can be used to predict the topics for the live data. This paper demonstrates the modelling of user based interest based recommendation system to provide relevant articles to the users.*

Keywords *- Latent Dirichlet Allocation, Event Detection, User Experience*

I. Introduction

Topic modelling is the way to analyse the large unstructured data and categorize them into specific topics. The topics are built based on the recurring pattern of co-occurring words across the documents. The topic modelling is more or less equivalent to clustering the data of similar category. In the process of topic modelling, the topics are created and associated with the list of words that describes the topic. Topic modelling helps to understand the latent structure of the document collection. This modelling technique comes in handy when there is a large scale of documents need to process.

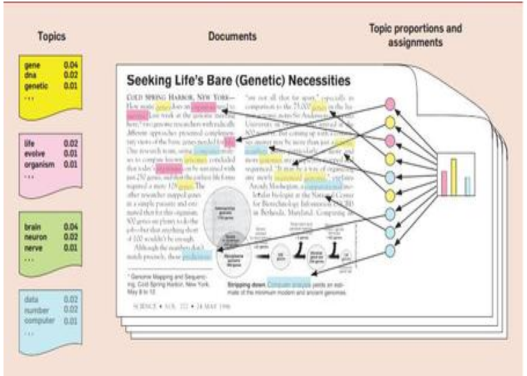


Figure 1: "Probabilistic Topic Models"

There are various Open source topic modelling tools available such as TMVE (Topic Model Visualization Engine), HDP (Hierarchical Dirichlet processes), CTR (Collaborative modelling for the recommendation), DTM (Dynamic topic models and the influence model). These tools look through a corpus for these clusters of words and groups them together by a process of similarity. A good topic model should provide the cluster of words which by themselves able to give a label. One of the most popular technique to perform topic modelling is by using **L**atent **D**irichlet **A**llocation

II. Related Works

Latent **D**irichlet **A**llocation (LDA) is a topic model that generates topics based on word frequency from a set of documents. LDA is particularly useful for finding reasonably accurate mixtures of topics within a given document set. LDA algorithm uncovers the hidden thematic structure in document collections, which enhance the way to search, browse and summarize large archives of texts.

The basic assumption of LDA is that each document contains a mixture of different topics and cluster of words associated with the topics. Each "topic" can be understood as a collection of words that have different probabilities of appearance in passages discussing the topic. One topic might contain many occurrences of "Mobile" "Gorilla Display" "3G/4G," and "Network." Another might contain a lot of "Cakes" and "Buffet".

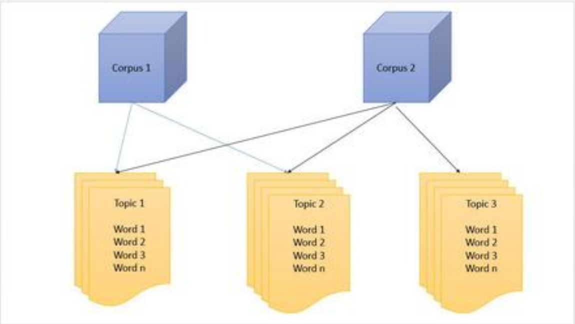


Figure 2: Sample topic modelling of corpus

A. LDA For Text Documents

Latent Dirichlet Allocation, as mentioned in [1], is a generative process that creates a set of documents. First, a corpus is mapped to two variables associated: α and β. α is the *k-dimensional* parameter of a Dirichlet distribution from which, for each document, we sample a mixture over k topics, called θ. Then, to each of the N word positions in the document a topic z_n is assigned by sampling from θ. When a word position's topic is known, the word w_n itself is selected according to $p(w_n|\beta, z_n)$, where β defines for each z_n a multinomial distribution over the vocabulary. In summary:

1. Choose $\theta \sim Dir(\alpha)$.
2. For each of the N_t word positions t_n:
   (a)Choose a topic $z_n \sim Multinomial(\theta)$.
   (b)Choose a word t_n from $p(t_n|z_n, \beta)$, a multinomial probability. conditioned on the topic z_n.

The sampling of N_t is usually left out of the equation

III. Problem Definition

Given a set of News Feeds **N** and Users **U,** need to find $\mathbf{N_{ui}}$ the $\subset$ **N.** where $\mathbf{N_{ui}}$ belongs to Topics in which the User **U** is interested, $\mathbf{T_{ui}}$ which is a $\subset$ **T**, the overallsuperset of topics.

IV. Preparing For Topic Modelling

A. Large Corpus

To perform the topic modelling, one needs a large document with the numbers of features and a larger set of documents, at least 1000 documents. To use the documents in various tools it has to undergo various pre-processing steps. The basic pre-processing steps can be tokenizing, removing stop words, tripping out the punctuation and removing capitalization. Then the document needs to break down into a list of words to make them suitable for the algorithm's input.

B. Tools to Perform Topic Modelling

There are various open source tools available for the topic modelling. The major overhead in using these tools are creating the input documents and determine the number of topics. It is important to be aware that you need to train these tools. Topic modelling tools only return as many topics as you tell them to; it matters whether you specify 50, 5, or 500. There are various methods to determine the number of topics for the input data. One of the approach is to find the elbow point using K-Means.

C. Understanding the Results

Topic modelling can be easily done by various tools available these days, but one of the major overhead in using them is to extract the necessary information from the output generated by those tools. One of the easiest way to analyze the results are to create a visual representation of the results. Topic modelling tools are fallible, and if the algorithm isn't right, they can return some bizarre results

V. Implementation

LDA assumes documents are produced from a mixture of topics.Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place.

LDA is a matrix factorization technique. In vector space, any corpus (collection of documents) can be represented as a document-term matrix. The following matrix shows a corpus of N documents **D1, D2, D3 … Dn** and vocabulary size of M words **W1, W2 ... Wm**. The value of **i, j** cell gives the frequency count of word **Wj** in Document **Di**.

	W1	W2	W3
D1	3	0	1
D2	7	9	5
D3	0	6	2

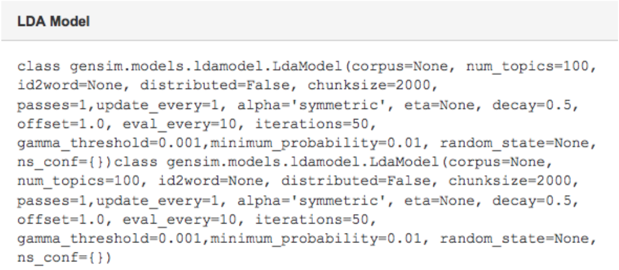
Table 1: Term Frequency Matrix

[2]Now the LDA algorithm will construct the Document Topic matrix and Topic - Term matrix from the Document Term matrix where the **N** number of documents has **K** number of topics associated with **M** number of vocabularies.

Once the basic model matrices are constructed the LDA now scans through each and every word in a corpus and try to adjust the topic **K** with the probability **P.** After numbers of iterations the algorithm will produce a stable model where document topic and topic term distributions are fairly good. This is the convergence point of LDA.

A. LDA with Python

Gensim provides LDA package for python. This module allows both LDA model estimation from a training corpus and inference of topic distribution on new, unseen documents. The model can also be updated with new documents for online training.



LDA Model

```
class gensim.models.ldamodel.LdaModel(corpus=None, num_topics=100,
id2word=None, distributed=False, chunksize=2000,
passes=1,update_every=1, alpha='symmetric', eta=None, decay=0.5,
offset=1.0, eval_every=10, iterations=50,
gamma_threshold=0.001,minimum_probability=0.01, random_state=None,
ns_conf={})class gensim.models.ldamodel.LdaModel(corpus=None,
num_topics=100, id2word=None, distributed=False, chunksize=2000,
passes=1,update_every=1, alpha='symmetric', eta=None, decay=0.5,
offset=1.0, eval_every=10, iterations=50,
gamma_threshold=0.001,minimum_probability=0.01, random_state=None,
ns_conf={})
```

Figure 3: LDA Model and Parameters

B. *Training Document Preparation*

The most difficult step in any ML model creation is to read the training documents. The documents need to be pre-processed to remove all the unnecessary and unwanted data. [3]This data cleaning is one of the most crucial steps which decides the quality of the topics that the algorithm will generate. The input documents can be fetched from various sources such as a database or even a file system.

C. *Document Pre-Processing*

Most common document pre-processing steps are Tokenizing, Stop-Words filtering and stemming.

1) *Tokenizing*

Tokenization segments a document into its atomic elements.

2) *Stopping*

Every document contains almost 30% of stop words such as articles, prepositions which are not a useful feature and they need to be cleaned up

3) *Stemming*

[4]Stemming words is another common NLP technique to reduce topically similar words to their root. For example, “stemming,” “stemmer,” Importing document from file System Tokenizing Stopping “stemmed,” all have similar meanings; stemming reduces those terms to “stem.” This is important for topic modelling, which would otherwise view those terms as separate entities and reduce their importance in the model. The most commonly used stemming algorithm was **PORTER STEMMING**. There are other algorithms like snowball stemming.

D. *Constructing Document - Term Matrix*

Once the done with pre-processing now it is time to convert them into document term matrix. This process will loop through each and every document and converts each string into an integer value and also collects the count of words in the document.

E. *Construct the Model*

[5]Now let us use the LDA module provided by the Gensim to construct the model. The module requires three mandatory arguments they are, Corpus-"Input documents", num_topics-"determine how many topics should be generated", id2word - "dictionary used in previous steps to map ids to strings". The user can also specify the number of times the algorithm process the input documents. More the number of passes more the time it will take to construct the model.

F. *Visualizing Topic Distribution*

The topic distribution of a model can be visualized with the top words belongs to the particular topic and the probability of the word distribution in that particular topic.

G. *Save Load and Update Stable Model*

Once the iteration, the number of topics and passes are stable user can save the model for the later topic prediction of live data. The saved model can be loaded at any point of the time and used for the topic modelling. The user can even update the model with the live data on the fly.

VI. Evaluation

A. *Dataset*

The dataset used was collected over a period of two years. Data set consist of News collected from approximately 10000 RSS and the user interest in the articles. There were over 10 Lakh articles with more than 50 Million user interests.

	2016	2017
News Articles	250000	280000
Press Articles	345000	475000

Table 2: Document count from different source for training

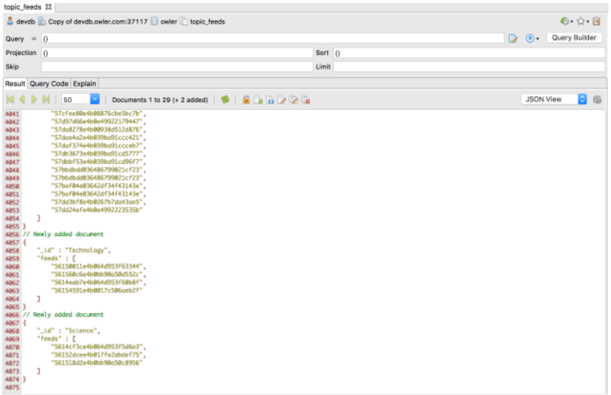
	2016	2017
User Count	1.2 Million	3 Million
User Interests	20 Million	30 Million

Table 3: User count and the number of cumulative interests shown by users

The user interest is gathered based on the articles viewed by each user and the number of times the articles was viewed. This gives the preliminary data about the topics that any particular user interested.

B. Results

The model is used to classify the articles across 170 topics and the incoming new articles are fed into the model to find the topic it belongs.



Query {}

Projection {}

Sort {}

Skip

Limit

50

Documents 1 to 29 (+ 2 added)

```
            "57cfee80e4b08876cbe5bc7b",
            "57d97d66e4b0e49922179447",
            "57da8278e4b00938d512d876",
            "57dae4a2e4b039ba91ccc421",
            "57daf374e4b039ba91ccceb7",
            "57db3673e4b039ba91cd5777",
            "57dbbf53e4b039ba91cd96f7",
            "57bbdbdd036486799021cf23",
            "57bbdbdd036486799021cf23",
            "57baf04e03642df34f43143e",
            "57baf04e03642df34f43143e",
            "57dd3bf8e4b0267b7da43ae5",
            "57dd24efe4b0e4992223535b"
        ]
}
// Newly added document
{
        "_id" : "Technology",
        "feeds" : [
            "56150011e4b064d953f63344",
            "561560c6e4b0bb90a50d552c",
            "5614eab7e4b064d953f60b8f",
            "56154591e4b0017c506aeb2f"
        ]
}
// Newly added document
{
        "_id" : "Science",
        "feeds" : [
            "5614cf3ce4b064d953f5d6a3",
            "56152dcee4b017fe2abdef75",
            "561518d2e4b0bb90a50c8956"
        ]
}
```

Figure 4: Feeds classified into various topics

Then the user input is given to the LDA topic modelling to determine the topics that a particular user is interested. The constant feedback is given back to the model so that the model will evolve over period of time.

While presenting the articles in the topic interest to the users there is a 30-percentage increase in the user engagement. When the user is provided with 15 articles it is more likely that the user reads 8-9 articles when the articles are in the topics that a user interested. This shows the high user engagement when the presentation of the articles are customized to the users.

VII. CONCLUSIONS

This paper proposes an algorithm based on the topic modelling for the recommendation of the articles to the users based on the both the topic that an article falls into and the topics that a particular user will be interested. Which increases the user engagement when the articles are recommended based on the topics they are interested.

VIII. REFERENCE

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3:993–1022, 2003.

[2] Wang, Chong, and David M. Blei. "Collaborative topic modelling for recommending scientific articles." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 11*, 2011, doi:10.1145/2020408.

[3] Andrzejewski, David, et al. "Incorporating domain knowledge into topic modelling via Dirichlet Forest priors." *Proceedings of the 26th Annual International Conference on Machine Learning - ICML 09*, 2009, doi:10.1145/1553374.1553378.

[4] Asuncion, Hazeline U., et al. "Software traceability with topic modelling." Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE 10, 2010, doi:10.1145/1806799.1806817.

[5] Tang, Jie, et al. "A Topic Modelling Approach and Its Integration into the Random Walk Framework for Academic Search." *2008 Eighth IEEE International Conference on Data Mining*, 2008, doi:10.1109/icdm.2008.71.