iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** VI **Month of publication:** June 2019

DOI: http://doi.org/10.22214/ijraset.2019.6334

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

A Comparative Study of NLP Topic Modeling Methods and Tools

Dr. Shivkumar Goel[1], Miss. Priyanka Devasthali[2]
[1]*Deputy Head,* [2]*Master of Computer Applications Department, Vivekanand Education Society's Institute of Technology (V.E.S.I.T), Mumbai, India*

Abstract: ***Analytics today is all about obtaining 'information' from data. Text mining techniques can rapidly gain valuable knowledge and insights from a large amount of unstructured data which is obtained from digital text-based datasets such as web pages, online documents, blogs, articles and emails. Topic modeling is a powerful form of text mining which can be used to extract the data and fetch the information that we are looking for. Topic models represent documents as a 'Bag-of-words' model without taking into consideration the order in which words appear. This paper is meant to study the comparison between four methods which come under the area of Topic Modeling. These methods are Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA). It is also meant to discuss tools available for Topic Modelling- Natural Language Toolkit (NLTK), Gensim and Mallet.***
Keywords: Analytics, Text Mining, Natural Language Processing, Topic Modeling, NLTK

I. INTRODUCTION

In today's world, where technology has taken over every aspect of life and is called a digital era, the web gets flooded with extremely large amounts of data generated every minute of every day. All the more, the addition of new data makes it difficult to access any relevant information we need. Hence, we need proper tools or methods to classify, organize and analyze this large volume of data.

Topic modelling provides us with methods to organize, analyze and summarize large sets of textual data. Its main uses include:

A. Finding out latent patterns that are present across the collection of data documents
B. Interpret documents according to the explored topics
C. Using these annotations to categorize, summarize and search the required texts

Topic modeling is an important area of data (text) mining. A topic model is a probabilistic model that finds out the main topics in a collection of data called as corpus. The basic concept is to consider the data or documents as a collection of topics in the topic model, and each topic is considered as a probability distribution of the words. Each topic in itself is viewed as a mixture of words, and each document can be viewed as a collection of topics with different probabilities depending on the frequency that those terms appear.
Topic Modelling is distinctive of rule-based text mining strategies that make use of regular expressions or dictionary based keyword searching routines. It is an unsupervised approach used for detecting the mass of words (called "topics") in large clusters of texts. Topics can be defined as "a repeating pattern of co-occurring terms in a corpus" [1].
An appropriate topic model should have the results as – "degree", "knowledge", "learning" and "college" for a topic – Education, and "computer", "software", "algorithms", "robotics" for a topic – "Technology".
Topic modeling for information retrieval these days has gained importance and has proved good performance in a number of tasks. It facilitates understanding, organizing and summarizing huge text datasets. A comparative study of various topic modeling approaches, including Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA) is presented in this paper. A number of tools are available for performing NLP. This paper discusses three of those- Natural Language Toolkit (NLTK), Gensim and Mallet.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

II. LITERATURE REVIEW

Due to recent advancements in the area of natural language processing, there has been constant and rigorous research going on all different fields of NLP. A number of articles and research papers have been taken into consideration while studying for this research paper.

Formally, we define the following terms:

1) A word is a discrete unit of data defined to be an item from a vocabulary indexed by {1,..,V}. We Using superscripts to denote components, the v^{th} word in the vocabulary is represented by a V-vector w such that wv=1 and wu=0 for $u \neq v$

2) A document is a sequence of N words denoted by $w = (w_1, w_2,..., w_N)$, where w_N is the n^{th} word in the sequence.

3) A corpus is a collection of M documents denoted by $D = \{ w_1, w_2,... ,w_M\}$. [2]

A. Text Analytics/ Text Mining

The process of exploring and investigating large volumes of written (text) resources to generate relevant information and to alter the unstructured text to structured data for use in advanced analysis is called as Text Analytics/ Text Mining. Text mining finds out various specifics, associations and assertions that would otherwise be left unseen in the huge pile of big data. These facts are extracted and converted into structured data, for data analysis, data visualization via graphs, html tables, charts, further having integration with structured data , and in the end includes refinement using machine learning (ML) algorithms [3].

Unstructured data is ubiquitous. The amount of unstructured text data is growing rapidly, and Computer World magazine declares that unstructured information might be more than 70%-80% of all data in organizations [4]. Hence, text mining is relevant to enable the effective and efficient use of huge quantities of text.

Text mining vs. Traditional Word Search

Considering a large number of documents and data, a traditional keyword search will find all the documents that contain the keywords specified by us and hence we need to know beforehand whether the documents provided by us are relevant to our research area. That works well enough if we know that those documents actually contain relevant information. However, text mining software would read the documents and analyze them on our behalf, understanding the actual meaning due to the sophisticated techniques developed in the area of Natural Language Processing (NLP).

B. Natural Language Processing (NLP)

NLP is an umbrella term that includes a wide range of methods that can be used to involuntarily evaluate a large capacity of text data. Some of these methods are "supervised" i.e. they require manual intervention of researchers by first classifying a sample set of documents and then use algorithms to “learn” word associations to apply the same manner to a larger collection of documents. The unsupervised methods don't require manual intervention of classification of training data. An observation of how words are used in the documents is done. These algorithms then pick up patterns and provide an approximation of what the documents convey with only small amount of the researcher’s indirect supervision. A topic model is one of the most popular forms of unsupervised learning for text analysis.

C. Topic Model

In natural language processing, a **topic model** is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents [5].

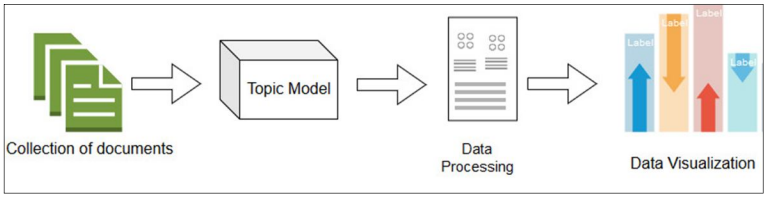


Fig.1.1 A general idea of a topic model

In a set of documents related to certain topics, one would expect specific words to be present in the document more or less frequently: "legislature" and "parliament" will be present more often in data about politics, "teaspoon" and "chop" will be present in cookbook documents, whereas "the", “it”, “on” and "is" will be present equally in both. Such words which do not add any value to

the topic recognition part of natural language processing are called as stopwords. These stopwords are removed from the data while cleaning and preprocessing it to derive topics from that data.

Multiple topics in different proportions are present in different documents.

All topic models are based on two basic assumptions:

1) Each document consists of a mixture of topics, and

2) Each topic consists of a collection of words.

In other words, topic models are built around the idea that the semantics of our document are actually being governed by some hidden, or "latent," variables that we are not observing. As a result, the goal of topic modelling is to uncover these latent variables — *topics* — that shape the meaning of our document and corpus [7].

D. How Topic Models Work

Every topic modeling algorithm has the hypothesis that the investigatory documents consist of a fixed number of topics. The model then examines the underlying words and their structure and attempts to find the groups of words that best "fit" the corpus based on that constraint. At the end, it generates two output tables: the term-topic matrix, which breaks topics down in terms of their word components, and the document-topic matrix, which describes documents in terms of their topics. Depending on the particular algorithm that you use, a word may be assigned to multiple topics in differing proportions, or assigned to a single topic completely.

In simple words, the term-topic matrix will give us a probability of how many times a word has occurred in a document. Since the raw output of numbers would be difficult to interpret, researchers often sort through the words for each topic and pick out the most distinctive words on the basis of their importance. This gives us a better sense of what content is present in our documents and identifies the theme of the documents.

In document-topic matrix, the topic model identifies how much part of the document covers a particular topic. For example, a document may comprise of 70% of topic 1 and 15% each of topics 2 and 3. While using a model that describes the proportion of various topics in a document, researchers can analyze the topics as continuous variables, as well as consider them as discrete classifications. This can be done by setting a threshold value to determine if a document contains a topic or not.

E. Advantages And Limitations Of Topic Models

Topic models have become a convenient tool for quantitative text analysis. Topic models can be much more beneficial than simple word search or dictionary-based strategies depending upon the application. Topic models give optimal results when used on data that is not brief, such as tweets, and that has a uniform composition.

Likewise, topic models have several valid limitations. To begin with, the term "topic" is somewhat unclear, and topic models will not be able to classify texts that have slight differences in the meaning of words that form the topic. Second, topic models can be misused easily if they are unfairly taken as a depiction of a purpose of the meaning of a text [8].

F. Applications of Topic Modeling:

Topic modeling helps in exploring huge amounts of text data, finding clusters of words, similarity between documents, and finding out abstract topics. Apart from this, topic modeling is also used in search engines wherein the search string is matched with the results.

III. TOPIC MODELING METHODS

This paper provides a description of three major methods of topic modeling as follows:

A. Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) is a technique in NLP, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are similar in meaning will occur in similar pieces of text, which is called as the distributional hypothesis.

A matrix containing word counts per paragraph is built from a large set of text data and a mathematical methodology called singular value decomposition (SVD) is used to reduce the number of rows while maintaining the similarity formation among columns. Paragraphs are then compared by taking the cosine of the angle between the two vectors (or the dot product between the normalizations of the two vectors) formed by any two columns. Values close to 1 represent very similar paragraphs while values close to 0 represent very dissimilar paragraphs [8].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

LSA is one of the most basic techniques in topic modeling. The basic idea is to take a term-document matrix and decompose it into

1) Document-topic matrix and

2) Topic-term matrix.

The first step is generating our document-term matrix. Given *a* documents and *b* words in our vocabulary, we can construct a *(a × b)* matrix *M* in which document is represented by rows words are represented by columns. In the simplest form of LSA, each entry can simply be a raw data of the number of times a particular word 'i' appeared in the i^{th} document. In practice, however, raw data counts do not work particularly well because they do not consider the importance of each word in the document. Therefore, LSA models replace the number counts in the document-term matrix with a tf-idf score. Tf-idf (term frequency-inverse document frequency), assigns a weight for term *j* in document *i* as follows:

$$w_{i,j} = t f_{i,j} x \log (N/df_j)$$

Where, $w_{i,j}$ = tf-idf score

$t f_{i,j}$ = occurances of term in a document

N = total no of documents

df_j = documents containing word

A term has a bigger weight when it occurs infrequently across the corpus but frequently across the document.

The document-term matrix *M* obtained by tf-idf will be very sparse, very noisy, and very redundant across its many dimensions. As a result, to find the few hidden topics that confine the relationships among the words and documents, dimensionality reduction is performed, using truncated SVD. SVD, or singular value decomposition, is a method in linear algebra that factorizes any matrix *M* into the product of 3 separate matrices:

$$M=U*S*V,$$

Where, *S* is a diagonal matrix of the singular values of *M*.

So, Singlular Value Decomposition provides us vectors for every document and term in our data. The length of each vector is taken as **k**. These vectors then use the cosine similarity method to find similar words and similar documents.

a) Limitations of LSA: LSA is quick and easy to implement and gives decent results, but it still has certain drawbacks:

i) Being a linear model, it does not do very well on non-linear datasets.

ii) LSA considers a Gaussian distribution of the terms in the documents, which may not be applicable to all scenarios.

iii) LSA involves SVD, which is computationally intensive and hard to update as new data comes up [9].

B. Probabilistic Latent Semantic Analysis (pLSA)

pLSA, or Probabilistic Latent Semantic Analysis, uses a probabilistic method instead of SVD to overcome the drawbacks of LSA. The basic idea is to find a probabilistic model with hidden topics that can create the data observed in the document-term matrix.

In particular, we want a model P(D,W) such that for any document *d* and word *w*, P(*d*,*w*) corresponds to that entry in the document-term matrix [7].

pLSA adds a probabilistic twist to the basic topic modeling basics:

1) Given a document d, topic z is present in that document with probability P(z|d)

2) Given a topic z, word w is drawn from z with probability P(w|z)

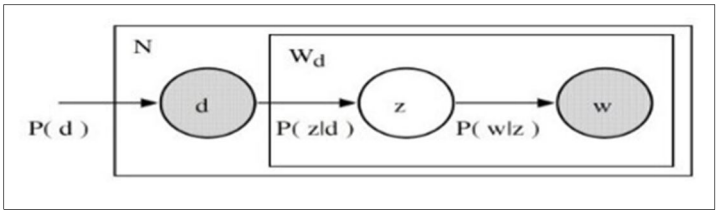


Fig. 2.1Probability distribution in pLSA [6]

The joint probability of a given document and word together is

$$P(D,W) = P(D) \sum P(Z|D)\, P(W|Z) - (i)$$

P(D) can be determined directly from our corpus. P(Z|D) and P(W|Z) are modeled as multinomial distributions, and can be trained using the expectation-maximization algorithm (EM).

P(D,W) can be equivalently parameterized using a different set of 3 parameters:

$$P(D,W) = \sum P(Z)P(D|Z)P(W|Z) - (ii)$$

In our first parameterization, we were starting with the document with P(d), and then generating the topic with P(z|d), and then generating the word with P(w|z). In *this* parameterization, we are starting with the topic with P(z), and then independently generating the document with P(d|z) and the word with P(w|z). [7]

pLSA adds a probabilistic treatment of topics and words on top of LSA. It is a far more flexible model and gives better empirical performance than LSA, but still has a few drawbacks:

a) Because we have no parameters to model P(D), we don't know how to assign probabilities to new documents

b) The number of parameters for pLSA grows linearly with the number of documents we have, so overfitting still remains a problem.

c) It is computationally expensive

d) The model is not humanly readable.

C. Latent Dirichlet Allocation (LDA)

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. 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 [2] In NLP, LDA is a generative statistical model which allows a number of exploratory observations to be understood by unobserved groups that define why some parts of the document are similar. For e.g., if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics [10].

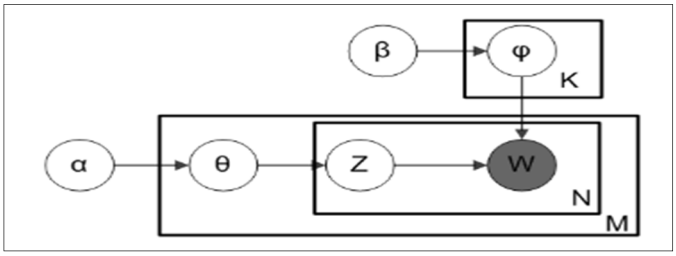


Fig. 3.1 Smoothed LDA [10]

Where,

α is the per-document topic distributions,

β is the per-topic word distribution,

θ is the topic distribution for document m,

φ is the word distribution for topic k,

z is the topic for the n-th word in document m, and

w is the specific word

1) The LDA is based upon two general assumptions:

a) Documents that have similar words usually have the same topic- e.g. Documents related to Healthcare will have similar correlated words like "vaccines", "medication", "diet", "surgery", etc.

b) Documents that have groups of words frequently occurring together usually have the same topic – i.e. if a specific set of words is frequently present together across various documents, then those documents may be of the same kind or may belong to a similar category.

2) Mathematically, the above two assumptions can be represented as:

a) Documents are probability distributions over latent topics

b) Topics are probability distributions over words [11]

3) Given the M number of documents, N number of words, and estimated *K* topics, LDA uses the information to output –
a) K number of topics,
b) psi, which is words distribution for each topic K, and
c) phi, which is topic distribution for document *i [12.]*
4) Methodology
a) Assume there are *k* topics across all of the documents
b) Distribute these *k* topics across document *m* (this distribution is known as α and can be symmetric or asymmetric, more on this later) by assigning each word a topic.
c) For each word *w* in document *m*, assume its topic is wrong but every other word is assigned the correct topic.
d) Probabilistically assign word *w* a topic based on two things: what topics are in document *m* how many times word *w* has been assigned a particular topic across all of the documents (this distribution is called as *β*)
e) Repeat this process a number of times for each document. [13]
5) Comparison of all Methods

Method	Characteristics and Applications
Latent Semantic Analysis (LSA)	▪ Quick and easy to implement ▪ Linear Model ▪ Distributional Hypothesis Applications – Document clustering in text analysis, recommender systems, building user profiles
Probabilistic Latent Semantic Analysis (pLSA)	▪ Far more flexible model ▪ Probabilistic model ▪ Produces better empirical results than LSA Applications – Object categorization, feature representation
Latent Dirichlet Allocation (LDA)	▪ Generative probabilistic model ▪ Iterative updating ▪ Uses logistic normal distribution to create relations between topics Applications – Bioinformatics, harmonic analysis for music and even object localization for images.

Table 1.1 Characteristics of Topic Modelling methods

Method	Limitations
Latent Semantic Analysis (LSA)	▪ Difficult to obtain the number of topics ▪ Does not work well in non linear distributions ▪ Less flexible
Probabilistic Latent Semantic Analysis (pLSA)	▪ Overfitting problem ▪ Computationally expensive ▪ Cannot determine probabilities of new documents.
Latent Dirichlet Allocation (LDA)	▪ Fixed k, the number of topics should be set in advance.

Table 1.2 Limitations of Topic Modelling methods

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

IV. TOOLS IN TOPIC MODELLING

This paper is meant to discuss three basic tools used for implementation of topic modeling:

A. Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) is a platform used for developing programs in Python programming language that work with human language data to enable natural language processing (NLP) applications.

NLTK implements simple interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum [14].

NLTK supports these important uses:

1) Tokenization
2) Identification of known entities
3) Displaying of a parse tree

B. Gensim

Gensim is a Natural Language Processing package that does 'Topic Modeling for Humans'.

It is a great package for processing texts, working with word vector models (such as Word2Vec, FastText etc) and for building topic models. An important advantage with gensim is that it allows us to handle large text files without having to load the entire file in memory.

It provides:

1) Tokenization by means of Dictonary object.
2) Document term matrix by using bag of words object (corpus)
3) TF-IDF matrix using models.TfidfModel() method.
4) Creation of bigrams and trigrams
5) Creation of topic models by using Lsi_Model(), Lda_Multicore() and Lda_Model() methods

A simple implementation of Topic Modelling may include the following steps:

a) Pre-processing:
i) *Tokenization:* The data is split into a number of sentences and the sentences are split into words. Lowercase the words and remove punctuation.
ii) Words that have fewer than 3 characters are removed.
iii) All stopwords are removed.
iv) *Words are Lemmatized:* Third person form words are modified to the first person and the verbs in past and future tenses are changed into the present tense. E.g.- caring is changed to care.
v) *Words are Stemmed:* Words are reduced to their root form.
b) Loading Gensim and NLTK libraries
c) Write a function to perform lemmatization.
d) Select a document to view the processing.

C. Mallet

The MALLET topic model package includes a rapid and very scalable implementation of Gibbs sampling, effective methods for document-topic hyperparameter optimization, and tools for inferring topics for new documents given trained models.

An implementation in Mallet Topic Modelling involves following steps:

1) *Importing Documents:* This imports the required documents as a single corpus by using 'import-dir' command.
2) *Building Topic Models:* Once documents are imported, the 'train-topics' command can be used to build the topic model.
3) *Hyperparameter Optimization Optimize-Interval [NUMBER]:* This option turns on hyperparameter optimization, which enables the model to better fit the data by allowing some topics to be more prominent than others [15].
4) Model Output
a) Output-model [FILENAME] This option designates a file to write a serialized MALLET topic trainer object. This type of output is suitable for pausing and restarting training but does not provide data that can easily be examined.

b) Output-state [FILENAME] This option generates a compressed text file holding the words in the corpus with their topic assignments. This file format can simply be parsed and managed by non-Java-based software.
c) Output-doc-topics [FILENAME] This option specifies a file to write the topic composition of documents.
d) Output-topic-keys [FILENAME] This file includes a "key" consisting of the topmost k words for each topic (where k is defined by the --num-top-words option). This output can help verify that the model is working as well as displaying results of the model. Also, this file reports the Dirichlet parameter of each topic.

If hyper parameter optimization is turned on, this number will be approximately proportional to the overall part of the collection assigned to a given topic [15].

5) *Topic Inference:* This allows us to build a topic inference tool based on the prevailing trained model.

V. CONCLUSION

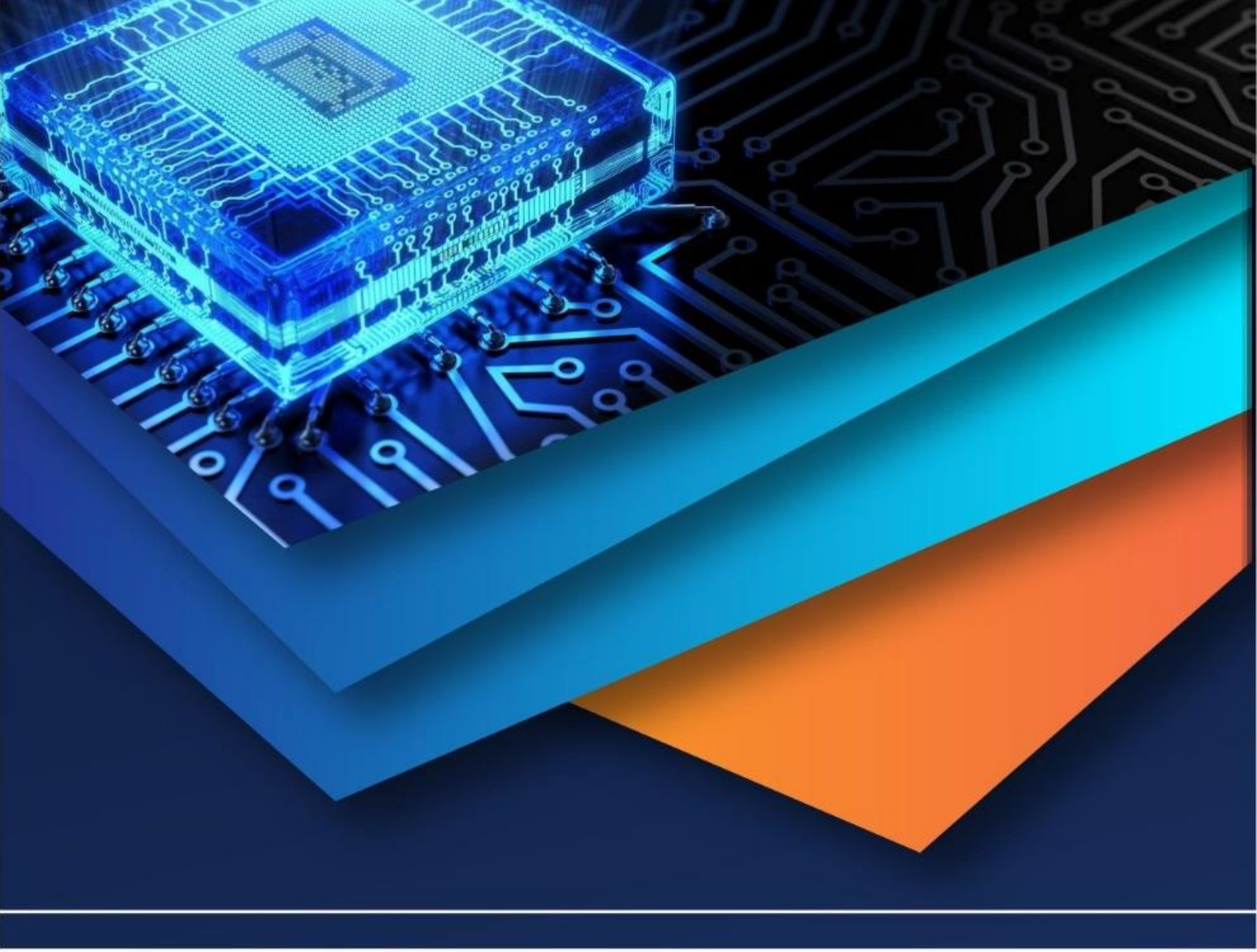
This research paper presents two categories of Topic modeling- Methods and Tools. The first category discusses three methods of topic modeling- LSA, pLSA and LDA. It specified the characteristics and theoretical backgrounds of these methods as well as its limitations and advantages over one another. The paper provides a high level view of each of these methods without going into specific details. LDA is the most widely used method of topic modeling. A number of methods, including both probabilistic and non probabilistic, were introduced before LDA for the sake of topic modeling and information gathering. Probabilistic models like pLSA and LDA improved on the previous non-probabilistic models. The second category provides a brief view of tools that can enable the use of topic modeling in various applications.

VI. ACKNOWLEDGMENT

This is to express our heartfelt gratitude to all students and staff of V.E.S.I.T to extend their support and cooperation to us that helped us to develop this research paper. It was their feedback that provided us better insights to do our research.

REFERENCES

[1] https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/
[2] David M. Blei, Andrew Y. Ng, Michael I. Jordan Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993-1022
[3] https://www.linguamatics.com/what-is-text-mining-nlp-machine-learning
[4] Chakraborty, Goutam. "Analysis of Unstructured Data: Applications of Text Analytics and Sentiment Mining" (PDF). SAS. Retrieved June 24, 2016.
[5] https://en.wikipedia.org/wiki/Topic_model.
[6] https://cbail.github.io/SICSS_Topic_Modeling.html.
[7] https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05
[8] Susan T. Dumais (2005). "Latent Semantic Analysis". Annual Review of Information Science and Technology. 38: 188–230. doi:10.1002/aris.1440380105.
[9] https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/
[10] https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
[11] https://stackabuse.com/python-for-nlp-topic-modeling/
[12] https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0
[13] https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd
[14] Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.
[15] http://mallet.cs.umass.edu/topics.php



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY