EJ-MATH, European Journal of Mathematics and Statistics
ISSN: 2736-5484

Keyword Extraction – Comparison of Latent Dirichlet Allocation and Latent Semantic Analysis

Bhuvaneshwari Kondeti, Jyothirani S. A, and Haragopal V. V

***Abstract* — The main aim of the present study is to compare the keywords extracted from abstracts and full length text of scientific research papers. In addition to that, here, we compare Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) to identify better performer for keyword extraction. This comparative study is divided into three levels, In the first level, scientific research articles on topics such as Indian Economic growth, GDP, Economic Slowdown etc. were collected and abstracts and full length text was extracted from the sources and pre-processed to remove the words and characters which were not useful to obtain the semantic structures or necessary patterns to make the meaningful corpus. In the second level, the pre-processed data were converted into a bag of words and numerical statistic TF-IDF (Term Frequency – Inverse Document Frequency) is used to assess how relevant a word is to a document in a corpus. In the third level, in order to study the feasibility of the Natural Language Processing (NLP) techniques, Latent Semantic analysis (LSA) and Latent Dirichlet Allocations (LDA) methods were applied over the resultant corpus.**

***Keywords* —Keyword Extraction, Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Natural language Processing.**

I. Introduction

The volume of text that is generated each day is dramatically increasing. This tremendous volume of text is mostly unstructured text cannot be simply processed and recognized by computers. Therefore, to discover useful patterns from this unstructured data, efficient and effective techniques and algorithms are required. Text mining or Text analysis is the task of extracting meaningful information from text, which has gained substantial attentions in the recent years due to increase in tremendous amount of text data [1]. Text mining techniques include categorization of text, summarization, topic detection, keyword extraction, search and retrieval, document clustering, etc. [2].

With the increasing volume of textual data particularly in research & news articles, keywords form an important factor as they provide a brief representation [3] of the article's content. Keywords also play a key role in finding the article from bibliographic databases, information retrieval systems and for search engine optimization. Keywords also help to categorize the article into the relevant topic or discipline. Conventional approaches of extracting keywords from the text data involve manual assignment of keywords based on the content and the authors' choice which involves lot of time & effort and also may not be accurate in terms of assigning the appropriate keywords. With the emergence of Natural Language Processing (NLP), keyword extraction has evolved into being effective as well as efficient [4]. Keyword extraction is the automated process of extracting the words and phrases that are most relevant to an input text [5].

LSA is one of the foundational techniques in topic modelling and Natural Language processing (NLP) that follows the same method as Singular Value Decomposition (SVD) [6]. LSA is a method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text data. LSA is an information retrieval technique [7] that analyses and identifies the patterns in unstructured text and the relationship between them. LSA uses document-term matrix [8] an input that describes the occurrence of group of terms in documents. It is a sparse matrix whose rows correspond to documents and whose columns correspond to terms.TF-IDF is an information retrieval technique that weighs a term frequency (TF) and its inverse document frequency (IDF) [9]. Each word has its respective TF and IDF scores. The product of the TF and IDF scores of a word is called the TFIDF weight of that word. LSA ultimately reformulates text data in terms of k latent (i.e. hidden) features, where k is less than n, the number of terms in the data.

Published on June 13, 2022.
B. Kondeti, Department of Statistics, Osmania University, India.
(corresponding e-mail: k.bhuvaneshwari7@gmail.com)
Jyothirani S. A, Rice Universi Department of Statistics, Osmania University, India.
(e-mail: jyothi2263@gmail.com)
Haragopal V.V, Department of Mathematics, BITS -Pilani, India.
(e-mail: haragopal_vajjha@yahoo.com).

Latent Dirichlet allocation (LDA) is a probabilistic model based on unsupervised learning, which assumes each document in a corpus like a random mixture of latent topics, and each topic has a probability distribution over all words in the vocabulary [10,12]. LDA is based on the idea that each document contains several hidden topics, each of which contains a collection of words related to the topic [10,11]. LDA discovers the latent topics Z from collection of documents D. For LDA, each document is a probability distribution over all words in the vocabulary. LDA model projects the documents in a topical embedding space, and generates a topic vector from a document, which can be used as the features of the document.

In this paper, we compare the two approaches of Natural Language Processing (NLP) i.e., Latent Semantic Allocation (LSA) and Latent Dirichlet Allocation (LDA), for keyword extraction on a dataset of scientific research papers relating to topics such as "Indian Economic Growth", GDP growth of India", "Economic Slowdown" etc.

II. Material and Methods

A. Data Collection

Data collection and preparation is the primary and most important step for research. In the present study, around 100 research articles on topics like "Indian Economic Growth", GDP growth of India", "Economic Slowdown" etc. were collected for past 5 years and the text from raw pdf files with field names "year", "abstract" and "full_text" was extracted to a csv file through python program.

The framework for this study is illustrated in Fig. 1. The analysis includes the following steps, i.e, Pre-processing, Exploratory analysis, Keyword extraction (The models used are LDA and LSA), Comparison of results i.e., key words extracted from abstracts and full text of the papers by LDA and LSA models.

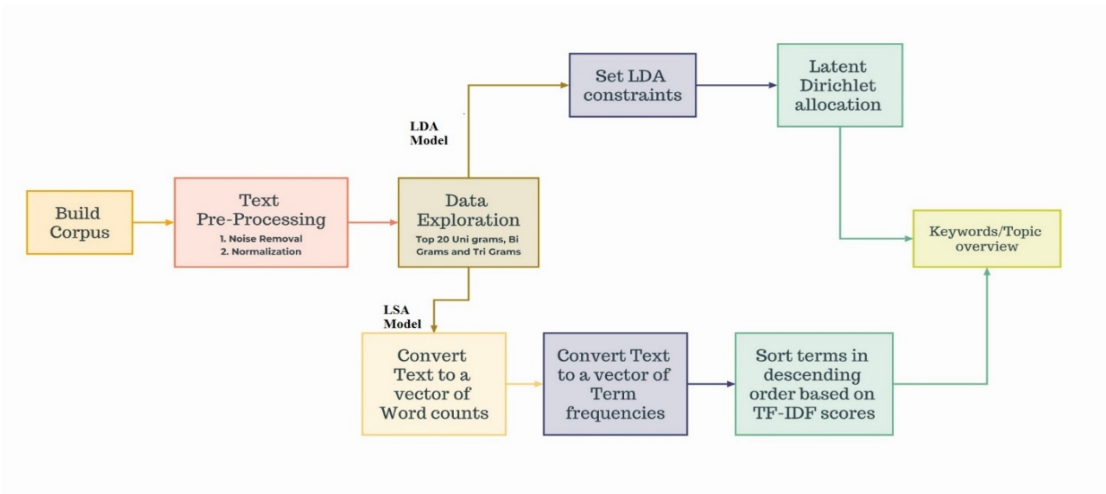

Fig. 1. Frame work for keyword Extraction by NLP models.

B. Pre-processing

The text data that we have is in raw form and can contain lot of noise and errors along with undesirable text due to which it will not give us results with accurate efficiency. In order to get better outcomes it is necessary to pre-process the text data for extracting the exact hidden information and makes it better to understand and analyse [13]. Text Pre-Processing involves removing of stop words, special characters and digits, punctuations, converting to lower case, stemming and lemmatization. This can be achieved by construction corpus object and importing *re* and *nltk* libraries in Python. The step by step approach of Pre-processing is explained in Fig.2.

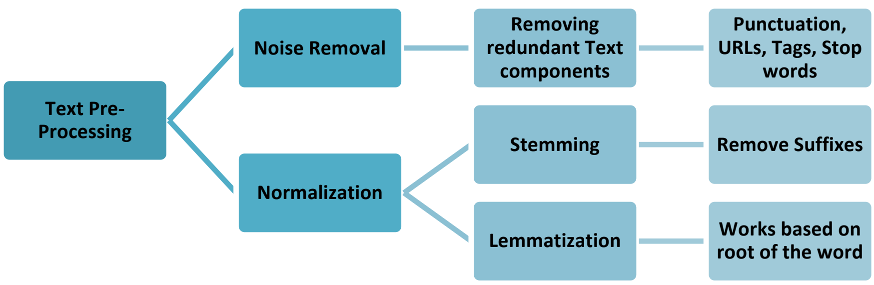

Fig. 2. Pre-processing steps of text data.

C. Data Exploration

Text in the corpus needs to be converted to a format that can be machine readable. There are 2 parts of this conversion i.e., a) Tokenisation and b) Vectorisation. For text preparation, Bag of words model is being used which considers frequencies of words rather than sequences. Then *CountVectoriser* class is used to tokenise the text and build a vocabulary of known words. We first create a variable "*cv*" of the *CountVectoriser* class, and then evoke *the fit_transform* function to learn and build the vocabulary.

D. Keyword Extraction

Keyword extraction is defined as the task which automatically recognizes a set of the terms or words that best describes the content of the document [14]. Extracting a small set of terms, composed of one or more words, from a single document is an important problem in Text Mining (TM), Information Retrieval (IR) and Natural Language Processing (NLP). The general framework of Keyword extraction is illustrated in Fig. 3.

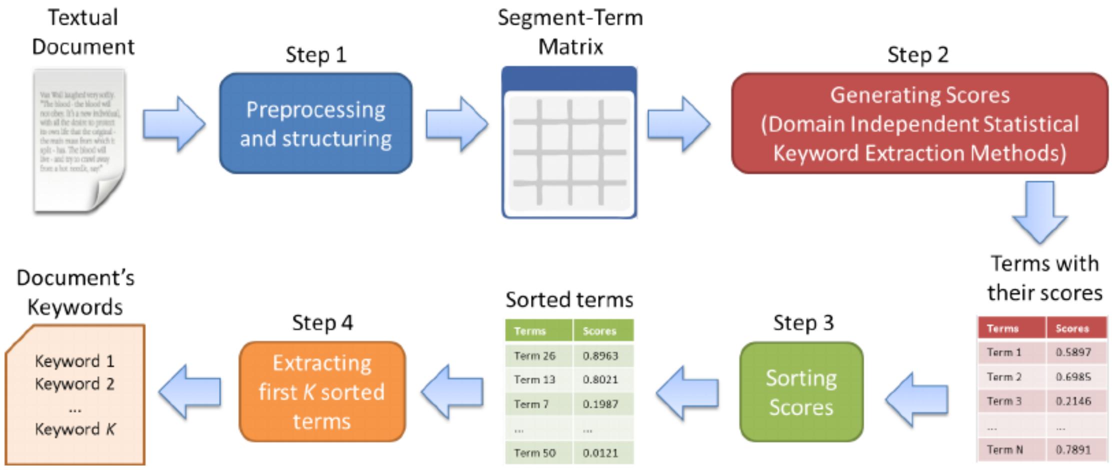

Fig. 3. Frame work for keyword extraction.

1) Latent Semantic Analysis

Latent Semantic Analysis (LSA) is also known as Latent Semantic Index (LSI). LSA uses bag of words (BoW) model [15], which results in a term-document matrix (occurrence of terms in a document). Here, Rows represent terms and columns represent documents. LSA learns latent (hidden) topics by performing a matrix decomposition on the document-term matrix using Singular value decomposition (SVD) [9]. The LSA approach can be illustrated as follows.

- Let there be m documents and n words in our vocabulary, and we can construct an $m \times n$ matrix A in which each row represents a document and each column represents a word.
- Each entry can simply be a raw count of the number of times the j-th word appeared in the i-th document.
- LSA models replaces the raw counts in the document-term matrix with a **tf-idf score**. *Tf-idf*, or *term frequency-inverse document frequency*, assigns a weight for term j in document I as shown in Fig. 4.

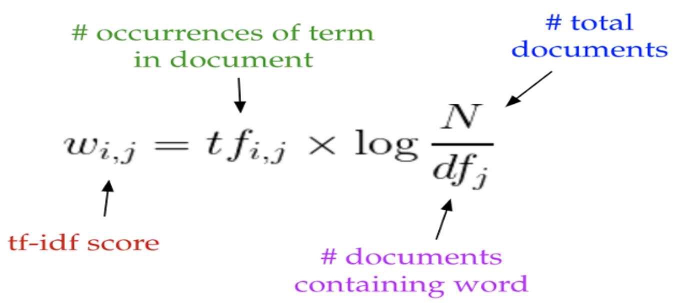

Fig. 4. Calculation of TF-IDF.

The term has a large weight when it occurs frequently across the document but infrequently across the *corpus*. The document term matrix A is very sparse, very noisy, and very redundant across its many dimensions. As a result, to find the few latent topics that capture the relationships among the words and documents, hence we have to perform dimensionality reduction on Matrix A using SVD.

Singular value decomposition (SVD), is a technique in linear algebra [16] 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.

Critically, truncated SVD reduces dimensionality by selecting only the *t* largest singular values, and only keeping the first *t* columns of *U* and *V*. In this case, *t* is a hyper parameter we can select and adjust to reflect the number of topics we want to find.

Now,

$$A \approx UtStVt^{T}$$

In this case, $U \in \mathbb{R}^{(m \times t)}$ emerges as our document-topic matrix, and $V \in \mathbb{R}^{(n \times t)}$ becomes our term-topic matrix. In both *U* and *V*, the columns correspond to one of our *t* topics. In *U,* rows represent document vectors expressed in terms of topics; in *V*, rows represent term vectors expressed in terms of topics. With these document vectors and term vectors, we can easily apply measures such as cosine similarity to evaluate

- the similarity of different documents
- the similarity of different words
- the similarity of terms (or "queries") and documents

In the present study, after performing the pre-processing steps, the bag of words model is being used to text preparation and a vector of word counts was created by the *CountVectoriser* class in Python. After that, the word counts were refined by using TF-IDF vectoriser. Large counts of certain common words may dilute the impact of more context specific words in the corpus. In order to overcome this, the TF-IDF vectoriser penalizes the words that appear many times across the document. Based on the TF-IDF scores, the words with the highest scores are extracted for both abstract and full length of a paper to get the keywords for a document.

2) Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a probabilistic model that extracts latent topics from a group of documents. The main idea is that the documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words [17]. LDA is termed as a Bayesian version of probabilistic latent semantic analysis method [18].

LDA assumes that the each document can be represented as a probabilistic distribution over latent topics, and again that topic distribution in all documents share a common Dirichlet prior.

Given a corpus D consisting of M documents, with document d having Nd words ($d \in 1,\ldots, M$), LDA models corpus 'D' according to the following generative process .

a) Select a multinomial distribution ϕ t for topic t ($t \in \{1,\ldots, T\}$) from a Dirichlet distribution with parameter β
b) Choose a multinomial distribution θ d for document d ($d \in \{1,\ldots, M\}$) from a Dirichlet distribution with parameter α.
c) For a word Wn ($n \in \{1,\ldots, N\ d\}$) in document d,
   i. Select a topic Zn from θd .
   ii. Select a word Wn from ϕ Zn.

The Graphic Model representation of LDA is shown in Fig. 5

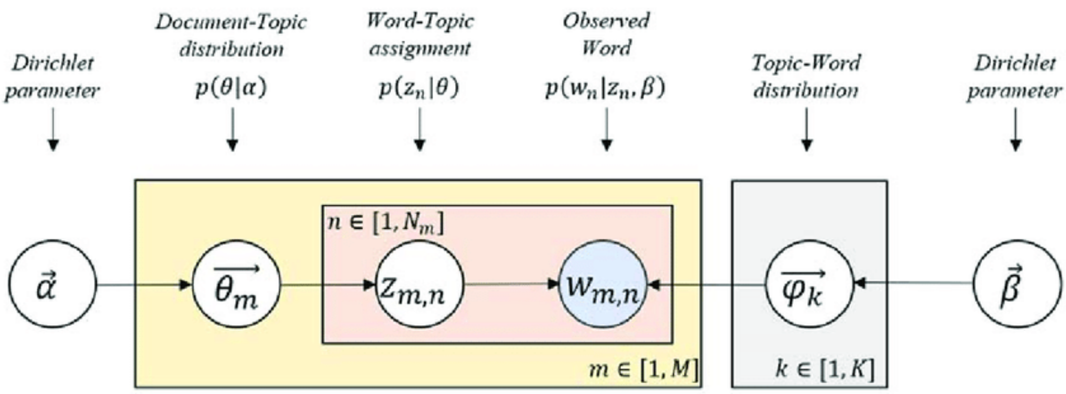

Fig. 5. Graphical model representation of latent dirichlet llocation.

In the present study, after performing the pre-processing steps, the topic model libraries and its dependencies were loaded in Python in order to perform LDA model for the corpus to determine the topics/keywords .Then, an optimal number of keywords (K) are determined for both abstracts and full texts of the papers in the dataset. LDA begins with random assignment of topics to each word and iteratively improves the assignment of topics to words through *Gibbs sampling.*

III. Results and Discussion

In this study we used 100 scientific research articles collected on topics such as Indian economy, Indian Economic Growth, GDP etc., for keyword extraction from both abstracts of the research papers and full length text of the papers. Further, we analysed two approaches of Topic modelling i.e., LDA and LSA for keyword extraction and compared the results obtained in both the models. In addition to that, we compared the results by increasing key words extracted from both abstracts and full length of the research papers.

After pre-processing steps, the wordcloud was created in order to visualize the corpus to get insights on the most frequently used words. The worldclouds using abstract and full length paper are shown in Fig. 6 and 7 respectively. Further, CountVectoriser class in python is used to visualise the top 20 unigrams bi-grams and tri-grams for both abstracts and full lengths papers which are shown in Table I and Table II.

Fig. 6.Wordcloud using abstracts.

Fig. 7. Wordcloud using full length papers.

TABLE I: TOP 20 UNI, BI & TRI GRAMS FOR ABSTRACTS

	Uni gram	Frequency	Bi-Gram	Frequency	Tri-Gram	Frequency
1	growth	85	economic growth	29	gross domestic product	7
2	india	77	indian economy	23	small scale industry	7
3	economic	68	gdp growth	10	trillion dollar economy	7
4	economy	67	growth rate	10	gdp growth rate	6
5	gdp	48	per cent	8	domestic product gdp	5
6	indian	40	small scale	8	public health expenditure	5
7	sector	38	tax revenue	7	gdp per caput	4
8	country	37	gross domestic	7	status legal tender	3
9	study	28	domestic product	7	global financial crisis	3
10	tax	27	per caput	7	economic growth country	3
11	paper	26	good service	7	india economic growth	3
12	impact	26	scale industry	7	product ranging traditional	3
13	policy	23	trillion dollar	7	ranging tradional high	3
14	rate	21	dollar economy	7	traditonal high tech	3
15	present	19	international trade	6	target proposed national	3
16	world	18	public health	6	proposed national manufacturing	3
17	covid	17	financial crisis	6	national manufacturing policy	3
18	per	17	developing economy	6	create million job	3
19	term	16	macro economic	6	million job end	3
20	variable	16	long run	6	job end well	3

After data exploration, Firstly, we used LSA approach for keyword extraction based on TF-IDF scores and extracted keywords (n= 5, 10, 15) from both abstracts and full length of the research papers. Then taking a sample of size 25 i.e., from 25 documents we compared keywords extracted from abstract and full length of paper and found common words. The results are shown in Table III.

Further, we analysed the above results using one way ANOVA to test the Number of common keywords from abstract and full length of paper are homogenous or not. The ANOVA results are shown in Fig. 8.

TABLE II: TOP 20 UNI, BI & TRI GRAMS FOR FULL LENGTH PAPERS

S.No	Uni gram	Frequency	Bi-Gram	Frequency	Tri-Gram	Frequency
1	growth	684	economic growth	156	public health expenditure	76
2	india	665	indian economy	140	small scale industry	57
3	economy	510	per cent	136	gdp growth rate	41
4	economic	462	growth rate	102	reserve bank india	36
5	gdp	439	gdp growth	91	post reform period	27
6	sector	372	per caput	89	foreign direct investment	25
7	country	313	health expenditure	87	gross domestic product	25
8	rate	301	public health	86	per caput public	24
9	indian	286	small scale	69	caput public health	24
10	per	279	scale industry	62	micro small medium	17
11	study	270	exchange rate	61	small medium enterprise	16
12	impact	266	tax revenue	54	gdp per caput	16
13	industry	244	international trade	53	per caput tax	16
14	government	231	black money	52	caput tax revenue	16
15	data	228	long run	52	global financial crisis	15
16	year	225	fiscal deficit	52	time series data	15
17	variable	204	long term	46	movement fundamental variable	14
18	period	199	bank india	38	good service tax	13
19	state	197	real estate	37	economic growth india	13
20	expenditure	197	reserve bank	37	internationa trade gdp	13

TABLE III: COMMON KEYWORDS EXTRACTED FROM ABSTRACT AND FULL LENGTH PAPER

S.No	Doc id	Keywords= 5	Keywords =10	Keywords = 15
1	1	1	1	2
2	23	2	3	4
3	32	1	2	3
4	57	1	1	3
5	73	2	4	5
6	45	2	3	6
7	18	3	7	7
8	79	2	2	4
9	56	2	2	3
10	31	2	4	5
11	20	3	3	7
12	88	2	4	5
13	36	0	0	1
14	42	1	5	5
15	97	3	5	6
16	69	0	1	1
17	8	3	4	4
18	15	0	0	0
19	52	2	3	3
20	84	2	5	6
21	26	3	6	7
22	4	3	5	6
23	61	1	2	6
24	90	3	4	7
25	44	2	3	4

Oneway

ANOVA

Common_keywords

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	81.947	2	40.973	14.697	<.001
Within Groups	200.720	72	2.788		
Total	282.667	74			

Post Hoc Tests

Multiple Comparisons

Dependent Variable: Common_keywords
LSD

(I) Keyowrds_Taken	(J) Keyowrds_Taken	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Keywords=5	Keyworkds=10	-1.320*	.472	.007	-2.26	-.38
	Keywords=15	-2.560*	.472	<.001	-3.50	-1.62
Keyworkds=10	Keywords=5	1.320*	.472	.007	.38	2.26
	Keywords=15	-1.240*	.472	.011	-2.18	-.30
Keywords=15	Keywords=5	2.560*	.472	<.001	1.62	3.50
	Keyworkds=10	1.240*	.472	.011	.30	2.18

*. The mean difference is significant at the 0.05 level.

Fig. 8. One way ANOVA and post hoc LSD for LSA approach.

Next, we proceeded with LDA approach for keyword extraction from abstracts and full length papers and found common keywords from abstract and full length of paper by extracting keywords (n =5,10,15) for each document. Similar to LSA approach here also we took sample of 25 documents and compared the common keywords extracted from both abstracts and full length papers respectively. The results are tabulated in Table IV.

Further, on analysing the above results in Table II using one way ANOVA to test the Number of common keywords from abstract and full length of paper are homogenous or not, we found that the keywords extracted are independent and are not homogenous of the documents. The ANOVA results are shown in Fig. 9.

From Fig. 8 & 9, it is clear that, the number of common keywords extracted from both abstract and full length paper are independent from different documents.

TABLE IV: COMMON KEYWORDS EXTRACTED FROM ABSTRACT AND FULL LENGTH PAPER

S.No	Doc id	Keywords= 5	Keywords =10	Keywords = 15
1	1	1	1	2
2	23	2	3	4
3	32	1	2	3
4	57	1	1	3
5	73	2	4	5
6	45	2	3	6
7	18	3	7	7
8	79	2	2	4
9	56	2	2	3
10	31	2	4	5
11	20	3	3	7
12	88	2	4	5
13	36	0	0	1
14	42	1	5	5
15	97	3	5	6
16	69	0	1	1
17	8	3	4	4
18	15	0	0	0
19	52	2	3	3
20	84	2	5	6
21	26	3	6	7
22	4	3	5	6
23	61	1	2	6
24	90	3	4	7
25	44	2	3	4

Oneway

ANOVA

Common_keywords

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	34.667	2	17.333	7.496	.001
Within Groups	166.480	72	2.312		
Total	201.147	74			

Post Hoc Tests

Multiple Comparisons

Dependent Variable: Common_keywords
LSD

(I) Keyowrds_Taken	(J) Keyowrds_Taken	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound
Keywords=5	Keyworkds=10	-1.200*	.430	.007	-2.06	-.34
	Keywords=15	-1.600*	.430	<.001	-2.46	-.74
Keyworkds=10	Keywords=5	1.200*	.430	.007	.34	2.06
	Keywords=15	-.400	.430	.355	-1.26	.46
Keywords=15	Keywords=5	1.600*	.430	<.001	.74	2.46
	Keyworkds=10	.400	.430	.355	-.46	1.26

*. The mean difference is significant at the 0.05 level.

Fig. 9. One way ANOVA and Post hoc LSD for LDA Approach.

IV. Conclusion

Keyword extraction plays a crucial role to find important keywords that can be used to represent the whole text. The key objective of our research is to emphasize on the two popular topic modelling techniques namely, Latent Dirichlet Allocation (LDA) and Latent semantic Analysis (LSA) for keyword extraction. We compared these two models i.e., LDA and LSA to identify a better performer for keyword extraction from abstracts and full length papers of scientific research articles based on topics such as Indian economy, GDP growth, Economic Slowdown etc. The work clearly conveys that, in both the models the keywords extracted are independent and not homogenous of the documents. Further, LSA approach of keyword extraction reveals that, as we increase the number of keywords, the number of common words from abstracts and full length papers are also increasing which means LSA is performing better when compared with LDA. In future work, by taking more challenging textual data we would further go for in-depth analysis and inspect the patterns that represent topics at a granular level by applying NLP methods.

V. References

[1] Tan AH. Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases.* 1999; 8: 65-70.

[2] Hussein H, Hafez A, Mathkour H. Selection criteria for text mining approaches. *Computers in Human Behavior*. 2015; 51: 729-733.

[3] Firoozeh N, Nazarenko A, Alizon F, Daille B. Keyword extraction: Issues and methods. *Natural Language Engineering.* 2020; 26(3): 259-291.

[4] Slobodan B, Mestrovic A, Martincic-Ipsic S. An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences.* 2015; 39(1): 1-20.

[5] Hong L. Text feature extraction based on deep learning: a review. *EURASIP Journal on Wireless Communications and Networking*. 2017; 1: 1-12.

[6] Merchant K, Pande Y. NLP Based Latent Semantic Analysis for Legal Text Summarization. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2018: 1803-1807.

[7] Hong T, Phan TT, Nguyen KP. An adaptive latent semantic analysis for text mining. *International Conference on System Science and Engineering (ICSSE).* 2017.

[8] Suzek, TO. Using latent semantic analysis for automated keyword extraction from large document corpora. *Turkish Journal of Electrical Engineering & Computer Sciences*. 2017; 25(3): 1784-1794.

[9] HM Mahedi, Sanyal F, Chaki D. A novel approach to extract important keywords from documents applying latent semantic analysis. *10th International Conference on Knowledge and Smart Technology (KST)*. 2018,

[10] Huaijin P, Jing W, Qiwei S. Improving Text Models with Latent Feature Vector Representations. *13th International Conference on Semantic Computing (ICSC)*. 2019: 154-157,

[11] Niu L, Dai X, Zhang J, Chen J. Topic2Vec: Learning distributed representations of topics. *2015 International Conference on Asian Language Processing (IALP)*. 2015: 193-196.

[12] Qi L. An Efficient Method for Text Classification Task. *Proceedings of the 2019 International Conference on Big Data Engineering*. 2019.

[13] Murugan A, Hill C, Nolan T. Text pre processing. *Practical Text Analytics*. Springer, Cham, 2019: 45-59.

[14] Onan A, Korukoglu S, Bulut H. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*. 2016; 57: 232-247.

[15] Wild F, Stahl C. Investigating Unstructured Texts with Latent Semantic Analysis. Advances in Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin, Heidelberg. 2007.

[16] Landauer, TK. *Handbook of latent semantic analysis*. Psychology Press, 2013.

[17] Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. *J Mach Learn Res.* 2003.

[18] Shams, Mohammadreza, Ahmad Baraani-Dastjerdi. Enriched LDA (ELDA): Combination of latent Dirichlet allocation with word co-occurrence analysis for aspect extraction. *Expert Systems with Applications*. 2017; 80: 136-146.