International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-8)

Research Article | **August 2017**

Similarity Detection Using Latent Semantic Analysis Algorithm

Priyanka R. Patil
PG Student, Department of Computer Engineering,
North Maharashtra University, Jalgaon,
Maharashtra, India

Shital A. Patil
Associate Professor, Department of Computer Engineering,
North Maharashtra University, Jalgaon,
Maharashtra, India

Abstract—

Similarity View is an application for visually comparing and exploring multiple models of text and collection of document. Friendbook finds ways of life of clients from client driven sensor information, measures the closeness of ways of life amongst clients, and prescribes companions to clients if their ways of life have high likeness. Roused by demonstrate a clients day by day life as life records, from their ways of life are separated by utilizing the Latent Dirichlet Allocation Algorithm. Manual techniques can't be utilized for checking research papers, as the doled out commentator may have lacking learning in the exploration disciplines. For different subjective views, causing possible misinterpretations. An urgent need for an effective and feasible approach to check the submitted research papers with support of automated software. A method like text mining method come to solve the problem of automatically checking the research papers semantically. The proposed method to finding the proper similarity of text from the collection of documents by using Latent Dirichlet Allocation (LDA) algorithm and Latent Semantic Analysis (LSA) with synonym algorithm which is used to find synonyms of text index wise by using the English wordnet dictionary, another algorithm is LSA without synonym used to find the similarity of text based on index. LSA with synonym rate of accuracy is greater when the synonym are consider for matching.

Keywords— Document Similarity, Wordnet, Text mining, Latent semantic analysis, Latent dirichlet allocation.

I. INTRODUCTION

Latent Semantic Analysis (LSA) [1] and Latent Dirichlet Allocation (LDA) [2] are two popular mathematical approaches to modeling textual data. Questions posed by algorithm developers and data analysts working with LSA and LDA models motivated to How closely do LSAs concepts correspond to LDAs topics? How comparable are the most significant terms in LSA ideas to the most imperative terms of relating LDA subjects? Are the same documents affiliated with matching concepts and topics? Do the report closeness diagrams delivered by the two calculations contain comparative record? LSA and LDA models, numerous other factor models of literary information, much in like manner. Both are use bag-of-words modeling, begin by transforming text corpora into term-document frequency matrices, reduce the high dimensional term spaces of textual data to a user-defined number of dimensions, produce weighted term lists for each concept or topic, produce concept or topic content weights for each document, and produce outputs used to compute document relationship measures. Yet despite these similarities, the two algorithms generate very different models. LSA uses vector index document (VID) to define a basis for a shared semantic vector space, in which the maximum variance across the data is captured for a fixed number of dimensions. In contrast, LDA utilizes regards each record as a blend of latent fundamental subjects, every theme is displayed as a blend of word probabilities from a vocabulary. Although LSA and LDA outputs can be used in similar ways, the output values represent entirely different quantities, with different ranges and meanings. LSA produces term idea and record idea connection matrix. LDA produces term-topic and document-topic matrices. Direct comparison and interpretation of similarities and differences between LSA and LDA models is an important challenge in understanding which model may be most appropriate for a given analysis task.

II. LITERATURE SURVEY

To assess existing methods model human semantic memory, compare generative probabilistic topic models with models of semantic spaces. Worried about a models capacity to extricate the list of a word arrangement so as to disambiguate terms have different implications in different context. Models are related to predicting related concepts. LSA and LDA are utilized as occurrences of these methodologies and looked at in word association task. The task of semantic similarity can be formulated at different levels of granularity ranging from word-to-word similarity to sentence-to-sentence similarity to document to-document similarity or a combination of these such as word-to-sentence or sentence-to-document similarity [1]. Mining is the way toward inferring for designs with in an organized or unstructured information. Different mining strategies out of which they different in the unique situation and kind of dataset is connected. The way toward removing data and information from unstructured content prompted the requirement for different mining procedures for valuable example disclosure. Data Mining (DM) and Text Mining (TM) is similarity both techniques “mine” large amounts of data,looking for meaningful patterns. A portion of the mining sorts are information, content, web, business Process and administration mining[2].

III. PROPOSED SOLUTION

The proposed system solely focuses on similarity detection in text from the collection of documents. Initially preprocessing of the data is done in which the input in the form of the text and is matching with the documents which are stored in dataset, also give the linkage of matching document in LDA algorithm. LSA algorithm with synonym algorithm the similarity of the sentences are based on index. Also giving the synonym list of that sentences by using the wordnet. Next is LSA without Synonym detecting similarity of sentence index wise and give linkage of matching document.

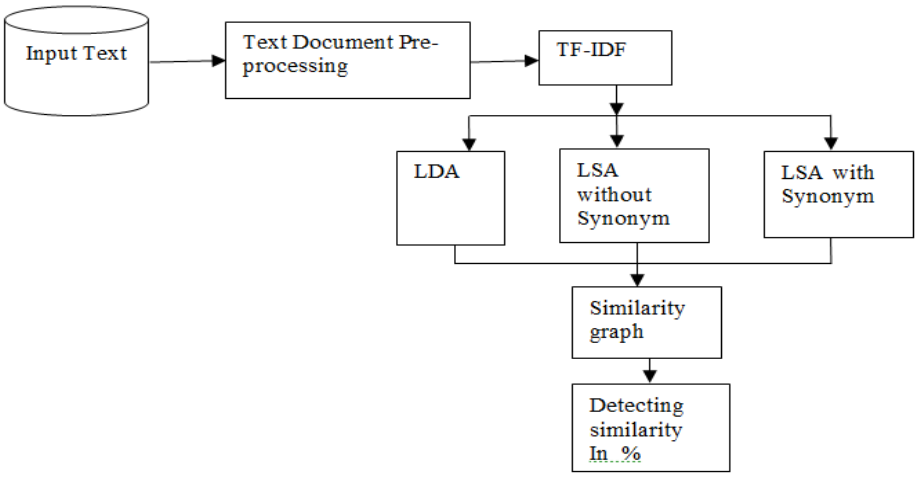

Fig. 1 Architecture of System

A. Approach

The proposed approach concentrates on preprocessing steps are utilized as a part of every one of the three calculations for likeness detection. First phase is about preprocessing in word tokenization, stop word removal common in LDA, LSA with synonym and LSA without synonym algorithms and synonyms matching from wordnet is included only in LSA with synonym

B. Preprocessing:

Preprocessing is the stage of converting the raw input into system desirable format. The raw input is the string or words which are directly given to the system; it needs normalized that into system acceptable format as shown in Fig 1.

1) *Tokenization:* The tokenization is required to separating the words from the sentences and produces the tokens.
2) *Stop word elimination:* The stop word are raw data or unnecessary data which have no any value for that purpose it has to be removed. It can be removed by using the Stop Word Removal Algorithm.
3) *Text Document Encoding*: On filtering content archives they are changed over into a feature vector. A progression utilizes TF-IDF calculation. Every token is relegated a weight, as far as recurrence (TF), mulling over a solitary research dad for each. IDF considers every one of the papers, scattered in the database and figures the opposite recurrence of the token showed up in all examination papers. So, TF is a local weighting function, while IDF is global weighting function. To find out the similarity within the document

$$Vi=TFi*\log(N/dFi) \quad (3.1)$$

Where; TFi= term frequency of feature word Vi,
N= No. of Document,
dFi= No. of document containing the word wi,

4) *Synonyms Matching*: Synonyms are extracted from wordnet . Feature vector is setup by estimating similarity function in Equation

$$Sim(w1,w2) = \begin{cases} 1 & w1,w2\ related \\ 0 & otherwise \end{cases} \quad (3.2)$$

A term matrix in Equation 3.3 is constructed to derive the semantic information content of S1 and S2.

let s1 U s2= S

S=$w_1;w_2;w_3;\ldots\ldots,w_n$, w_i= distinct &

$$s_i = \begin{cases} q1 & ; j_{11} \\ q2 & ; j_{22} \\ . & \\ . & \\ . & \\ q_n & ; j_{nn} \end{cases} \quad (3.3)$$

D(synonyms) si=s1,s2 s1=s2=1
Otherwise 0

Where; Q=query
j=sentence index document

$$Rs = (C/Sn) \times 100 \quad (3.4)$$

Where; Rs= result in percentage
C= Matching content
Sn=no. of sentences

C. LDA, LSA with Synonym And LSA without Synonym

In LDA Similarity detection can be done in LDA using the Preprocessing steps. Preprocessing is the stage of converting the document into sentences. Tokenizing the sentences or document into tokens and Stop words are removed.

In LSA with synonym after the preprocessing is complete the input sentences compare with dataset document by indexed wise if the sentence found on same index then give the similarities of sentence with synonym list of that sentence by using the English wordnet. Give the result how much percentage is match with document. Let, Match sentence with other sentence of document with index.

In LSA without synonym word calculation after the preprocessing is finished the info sentences is contrast and dataset record by ordered shrewd if sentence found on same list then give the similarities of sentence with link of matching document. Give the result how much percentage match with document.

D. Design

The LDA algorithm consists of Tokenization, Stop word removal. After the preprocessing step is complete LDA algorithm can be match similarity of input text file with the collection of document and giving list of matching document link.

Step by Step algorithm of LSA with Synonym is described in Algorithm 2. It consists of Tokenization, Stop word removal, and Synonyms matching algorithm for checking the synonyms word for indexed, it is extracted from synset word is in wordnet dictionary and match with synonym words. Giving list of Synonyms.

1. *Algorithm: LDA*

1) procedure Tokenization .
2) Require: Text query as input containing pairs of sentences
3) Input: txt file as input containing number of sentences
4) Output:array of words
5) repeat
6) tokenization
7) Remove special characters and symbols like ,""[].?/ etc.
8) until each word is uniquely identified
9) end procedure
10) procedure Stop word removal . Removing the stop words
11) Require: Tokenized arraylist tij from procedure 1
12) Input: txt file as input containing number of sentences
13) Output:array list of removed stop words
14) Store all stopwords in arraylist L
15) Compare tokenize arraylist tij to list L and remove that word by following,
16) For (all si in S) do
17) For 1 to j do
18) Extract tij from input file
19) If tij is in L then
20) Remove tij
21) End if
22) End for
23) Store other sentence words in new arraylist.
24) end procedure
25) procedure Similarity Detection . Matching input text with document
26) Input:txt file as a input
27) Output:Result in percentage based on matching content
28) Match Each sentence with document D
29) if Si==Q go to step 4
30) if sentence match with document Store document URL
31) increment counter
32) add D in Ld
33) Matching result in percentage
34) end procedure

2. *Algorithm: LSA with Synonym*

1) procedure Synonyms Wordnet Algorithm . Synonym matched by indexed
2) Input:txt file as a input
3) Output:Result in percentage based on matching content
4) Sentence words are pass it to the English wordnet Dictionary and check with synset words.

5) Get Synonyms words from wordnet.
6) Generate Synonym for each sentence
7) Read the sentence of query indexed wise(Si)
8) Read sentence of dataset document indexed wise(Sd).
9) If Si==Sd
10) Add in result Document
11) Else skip
12) Matching result in percentage
13) End procedure

3. *Algorithm: LSA Without Synonym*
1) procedure Similarity Detection by indexed . Matching input text with document
2) Input:txt file as a input
3) Output:Result in percentage based on matching content
4) read the sentence of query index wise(Si)
5) read sentence of dataset document index wise(Sd).
6) If Si==Sd
7) Add in result Document
8) else skip
9) Matching result in percentage
10) end procedure
11)

IV. CONCLUSIONS

A. Results

The collection of international journals are stored in database. Results are giving the url of that matching document. For Synonym generation used the English wordnet dictionary on proposed system.

1) F-Measure is defined as A measure that combines precision and recall is the harmonic mean of precision and recall as shown in Table 1 and Fig 2.

$$\text{F-Measure} = \frac{2(Precision * Recall)}{(Precision + Recall)}$$

Table1 Comparison of LDA, LSA with Synonym and LSA without Synonym Algorithm

Algorithms	Precision	Recall	F-Measure
LDA	68	85	75.55555556
LSA with Synonym	86.36	95	90.47419497
LSA without Synonym	75	90	81.81818182

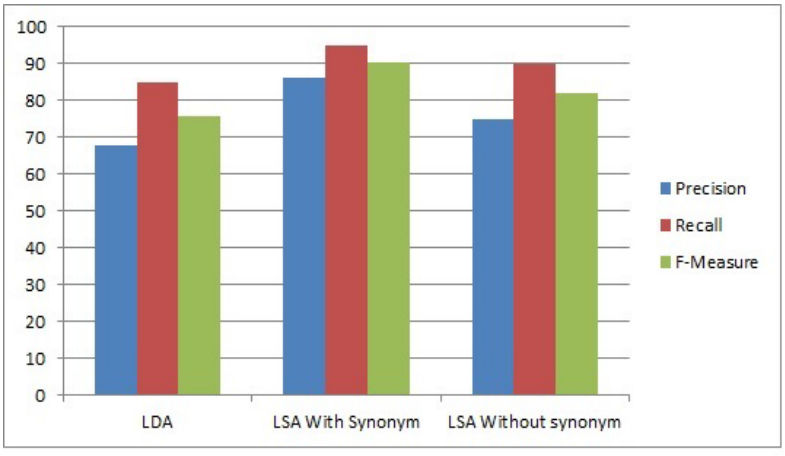

Fig. 2 Comparisons of LDA, LSA with Synonym and LSA without Synonym Algorithm

LSA with synonym, LSA without Synonyms and LDA models for document similarity. Algorithms are similar to the success of test similarity, semantic analysis document similarity is a promising direction. In future work to implement LSA algorithm in image retrieval and natural language processing. May be interesting part to implement proposed system on real-time applications to check the performance of system.

ACKNOWLEDGMENT

I would like to take this opportunity to express my deepest gratitude to respected Prof. Dr. Girish K. Patnaik, Head, Department of Computer Science and Engineering, S.S.B.T's College of Engineering and Technology, Jalgaon for his valuable advances and providing an opportunity to project and also providing comments and suggestions. I am truly

grateful to my project guide Mrs. Shital A. Patil, for her valuable guidance and encouragement. Her encouraging words went a long way in providing the patience and perseverance, which were needed to present this project thesis successfully. Also her true criticism towards technical issues provided me to concentrate on transparency of my project.

REFERENCES

[1] Niraula, Nobal and Banjade, Rajendra, *Experiments with semantic similarity measures based on lda and lsa*, in International Conference on Statistical Language and Speech Processing Springer, 2013, pp. 188-199.

[2] Vidhya, KA and Aghila, G, *Text mining process, techniques and tools: an overview*, in International Journal of Information Technology and Knowledge Management, 2010, pp. 613--622

[3] Balamurugan, R and Pushpa, Dr S, *A Review On Various Text Mining Techniques And Algorithms*, 2nd International Conference on recent innovations in science, engineering, and management JNU convention center, New Delhi, 22nd November, 2015.

[4] S.kanan, Gurusamy, Vairaprakash, *Preprocessing Techniques for Text Mining,* International Journal of Computer Science and Communication Network , 2014.

[5] Crossno, Patricia Joyce and Wilson, Andrew T and Dunlavy, Daniel M and Shead, Timothy M, *TopicView: Understanding Document Relationships Using Latent Dirichlet Allocation Models*, 23rd IEEE International Conference on Tools with Artificial Intelligence, 2011.

[6] Vishnu Priya and S.K.Sundaram, *A Latent Dirichlet Allocation Algorithm for Pattern-Based Topic Filtering*, Journal of Scientific Research, 2016.

[7] Crain, Steven P and Zhou, Ke and Yang, Shuang-Hong and Zha, Hongyuan, *Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond*, Journal of Mining text data",Springer, 2012,pp.129—161.

[8] Lee, Sangno and Baker, Jeff and Song, Jaeki and Wetherbe, James C, *An empirical comparison of four text mining methods*, 43rd Hawaii International Conference on System Sciences (HICSS), 2010. IEEE

[9] Sukanya, M and Biruntha, S, *Techniques on text mining*, IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2012

[10] Singh, Vikram and Saini, Balwinder, *An Effective Tokenization Algorithm For Information Retrieval Systems*, Journal of IEEE,2014.

[11] Balachandran, Vipin and Deepak, P and Khemani, Deepak *Interpretable and reconfigurable clustering of document datasets by deriving word-based rules*, Journal of Knowledge and information systems ,Springer 2012.

