Analysis of Topic modeling methods for efficient topic discovery and visualization

Ram Prasath Meganathan
B00851418
Faculty of Computer Science
Dalhousie University
Halifax, Canada
rm507817@dal.ca

Abstract—Collecting relevant and needed information from a large set of documents is a tedious and time-consuming process. To get the relevant output from documents, there are lot of mining techniques available. One of the most effective technique is topic modeling. This can be used to analyze patterns of input from customer feedback, extract features from job posts, Clustering and categorization. There are lot of topic modeling techniques available. We aim to compare few of those available models through experiment and use the results to find out the better model for performing topic modeling.

Keywords—topic modeling, clustering topics, visualization and evaluating documents

I. INTRODUCTION

News is a valuable source to know the current affairs and happenings around the world. It is communicated to the people through different mediums such as Newspapers, broadcasting, online videos. It plays a vital role in every field from politics till the fashion industry. The captured data is analyzed further to know the current trend of the industry as well as make future forecasting on which area would be generating more news content. Lots of media organizations are currently using their data to make such predictions. To start, they have to understand and identify the most spoken topic and focus their research over there. Organizing the topics into a cluster manually is an extremely difficult task as it would consume lot of time and efforts and it will not suitable for a fast-moving media industry. To save us the efforts we have a machine learning technique called Topic modeling which provides great way to organize news content and organize them into different topics. It uses unsupervised learning technique to cluster results. In the real world, it is used in many areas to identify patterns [1]. For illustration, Customer service industries use customer feedback to analyze and predicts patterns on the issues or complaints they will receive in the future. It works on the assumption that the document contains mixture of topics which can further be added by statistical distribution [1]. We have different algorithms to do a topic modeling on documents some of them are Latent Semantic Analysis, Latent Dirichlet Allocation, Hierarchical Dirichlet Process, Non-negative Matrix Factorization etc.

II. DATA RESOURCES

A. Online sources

We selected the dataset from the Kaggle website https://www.kaggle.com/akashram/topic-modeling-intro-implementation/data [4]. We have chosen the news headlines data since it has all the features of the big data. It is a daily growing data and scope of it continues to evolve and gives us better results on big data. The dataset contains the headlines of the articles of different categories as our problem addresses the need to perform topic analysis on a wide range of documents.

III. RELATED WORK ON AVAILABLE TOPIC MODELING METHODS

A. Latent Semantic Analysis (LSI)

This is one of the most common topic modeling techniques used. It makes use of statistical distributional hypothesis to identify the latent relationships in documents. It basically takes the whole document and forms a matrix of word frequencies. These word frequencies are formed by a frequency-based approach called tf-idf which is a combination of term frequency which is the number of occurrences of a particular word and the

inverse document frequency which is the number of times a word appears in a document. Multiplying these two matrices gives us the resultant matrix. The final resultant matrix can be further be decomposed using a term into three different matrices using a term called Singular value decomposition (SVD). It forms the Document topic matrix, term-topic matrix and the diagonal will be taken as S matrix [1][2].

Limitations: it doesn't consider the syntax and semantics of the text data that is passed as input.

B. Latent Dirichlet Allocation (LDA)

Like LSA, this is also a commonly used topic modeling algorithm. According to LDA, each document is topic of words and each topic is collection of words. Its main focus lies on the finding the topics involved in the corpus and group them together. It also operates under the assumption that each topic is a bag of words and it removes the most common stop words to help us identify a meaningful list of topics [2][3][4].

Assumptions: Order and the grammar of the words are not taken into account.

Limitations: we should specify how many topics are needed before hand.

P (word in topic) = P (topic document) * p (word w |topic)

C. Hierarchical Dirichlet Process (HDP)

It is a Bayesian procedure to cluster the data. It takes advantage of Dirichlet for each cluster as each Dirichlet groups follow a same distribution. Each topic is assumed as a cluster of topics and every word is fetched from that cluster. The uncertainty in the topic count is identified through the Dirichlet process. A base distribution is chosen to determine the achievable topics for the document from which the measurable number of topics is chosen from [3][4]

Limitations: This problem can be applied to a dataset where the number of topics is not determined at the beginning

D. Non-Negative Matrix Factorization (NMF)

It comprises of combination of algorithms where a matrix V is further computed to form two different matrices W and H. Each matrix containing only the positive. The absence of non-negative values makes the final matrix inspection easier [4].

It can be depicted as V= WH where V is the combination of column vectors with coefficients provided by W and H.

It differs from LDA by means of factorization of document-term matrix from which term-feature and feature document matrices are formed.

It can be formulated as follows V is factorized into n* t/ where t is the count of identified topics [12]

IV. EXPLORATORY DATA ANALYSIS

As part of the exploratory data analysis the given data set is explored, and the statistics of the data has been analyzed. This exploration is done using the pandas profiling report [7] and line plot libraries. Following are the screenshots of the results from the analysis that shows the stats of the analyzed dataset

Fig 1. below shows that there are four categories of data with 'e' category occupying the dominant distribution in the dataset. And Fig 2. represents the stats of the Title column with distinct count of values.

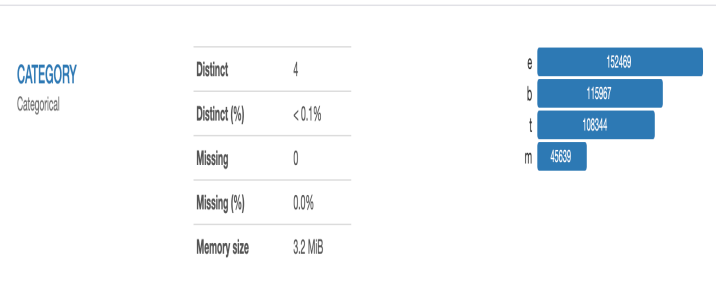


Fig 1. Category values involved in dataset

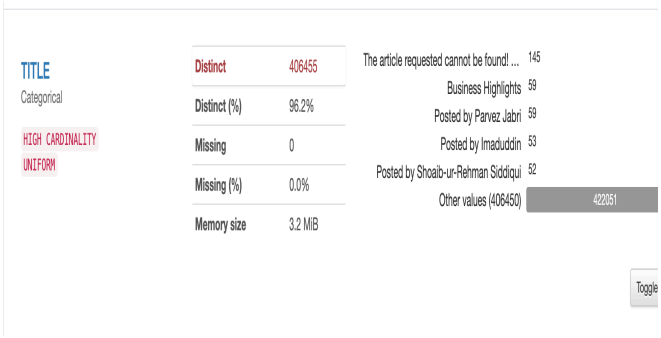


Fig 2. Column Title stats involved in dataset

Following is the total column count it with missing value counts for each column requiring the need of cleanup

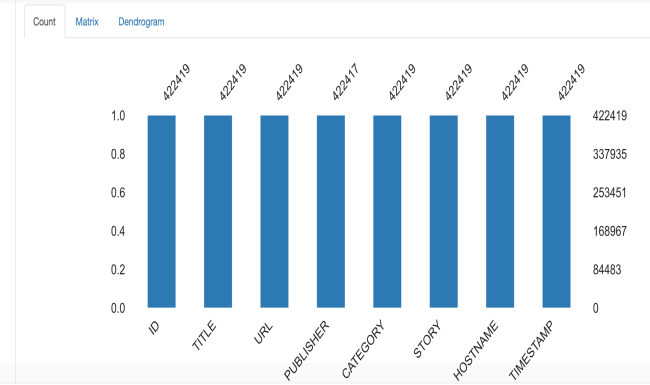


Fig 3. columns indicating missing values for columns in dataset.

The following word clouds are used to represent the values in the dataset title values in a unigram representation which shows the visual representation of the possible topics that we could get as a result from the dataset. Fig 4. denotes the word cloud of unigrams formed using the dataset.

Fig 4. Word cloud of unigrams present in the document.

Following is the word cloud of bi-grams present in the document which shows some more possible topic combinations. It is shown in Fig 5.

Fig 5. Word cloud of bigrams present in the document

Following is the plot of the dataset values category with the publisher which denotes the publishers to category ratio present in the dataset and also gives us an idea about the different types

of publishers available in the dataset. It is plotted using plotly library *[11]*

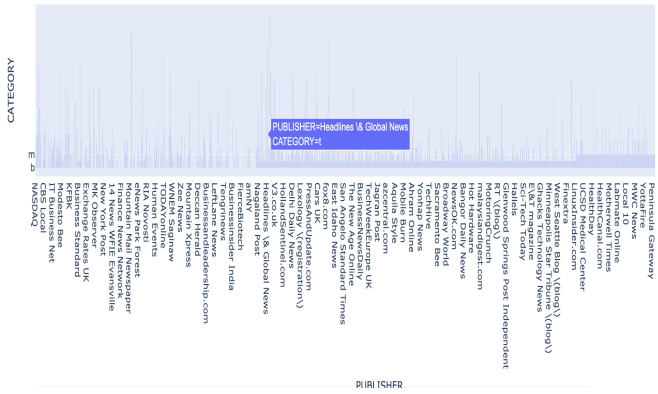


Fig 6. category to publisher plot

V. TOPIC MODEL METHODOLOGIES CHOSEN AND IMPLEMENTATION

We have chosen three available models for performing the topic modeling and showcase the results for a comprehensive comparison. They are Latent semantic Analysis, Latent Dirichlet Allocation and Hierarchical Dirichlet process. Fig 7. is the data flow diagram of the experimental setup to be performed on the dataset.

All chosen models undergo similar level of data preprocessing before they can be served as input to the topic model approaches. The output topics generated from each set is further plotted and their evaluation metrics are computed to derive the best model out of the three.

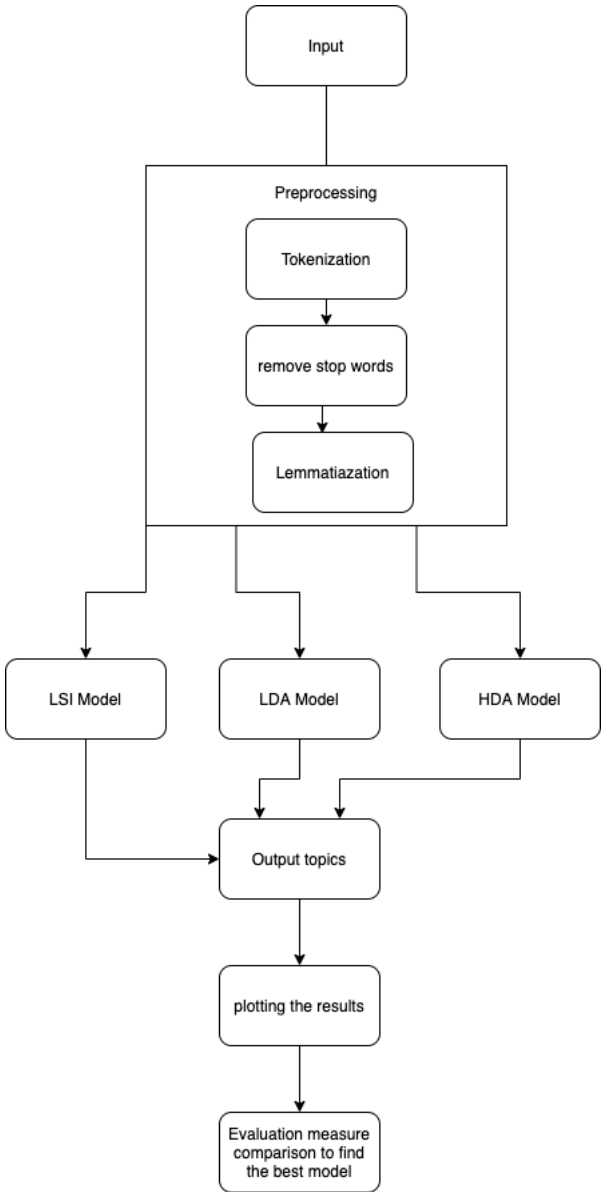


Fig 7. Data flow diagram of topic modeling comparison [16]

A. DATA PREPROCESSING

We have performed various preprocessing steps to clean and extract the needed values from the dataset. Through the exploratory data analysis, we identified the column having null values and removed those to have a refined dataset. Following are the preprocessing steps we have performed on the data model

1. Cleanup sentences

The entire document is cleaned using regular expression that removes the single quotes,

punctuations and other special characters from the sentences to make it ready for tokenization.

2. Tokenization

Tokenization is the process of tokenization the document to sentences or sentences to words. In this we approach, we follow a word tokenization approach performed by genism library which breaks the sentences to words for feeding to the stop word removal phase.

3. Stop words removal

The stop words were identified from the NLTK [10] corpus dataset and Scikit and they were merged together to form a new set of stop words that are to be removed from the input. After combining both the datasets total of 502 stop words were formed for further processing.

```
[['american',
  'apparel',
  'board',
  'refuses',
  'to',
  'meet',
  'with',
  'ousted',
  'ceo',
  'dov',
  'charney'],
 ['nabe',
  'economists',
  'say',
  'fed',
  'is',
```

Fig 8. Some of the Stop words used

4. Forming Bigrams and Trigrams

Bigram is combination of two words occurring adjacent to each other from a string of documents. Trigram is a combination of three words occurring together in a document. They are formed using the genism library and further passed as input for lemmatization.

5. Lemmatization

Lemmatization is the text normalization process which reduces the inflected word forms to ensure that the word that is formed is a word that belongs in the language. The reduced word which is formed as a result of the lemmatization is known as lemma. Following is the tabular form of stemming and lemmatization and Fig 9 shows the output of the dataset.

Word	Stemming	Lemmatization
was	be	wa
studies	studi	studi
studying	study	study

Table 1. Lemmatized and stemmed word difference table [18]

```
[['refuse', 'oust'],
 ['economist', 'say', 'right', 'track', 'survey'],
 ['print', 'make', 'commercial', 'ink'],
 ['target', 'profit', 'fall', 'breach', 'take', 'toll'],
 ['high', 'expect', 'growth', 'result', 'policy', 'easing'],
 ['car', 'ignition', 'problem'],
 ['crimea'],
 ['order', 'close', 'russian', 'court'],
 ['stock', 'close', 'pc'],
 ['keep', 'rate', 'hold', 'wait', 'stimulus', 'bite'],
 ['drug', 'ready', 'bid', 'four'],
 ['ca', 'business', 'summary'],
 ['strike', 'ground', 'flight'],
 ['unemployment', 'fall'],
 ['china', 'export', 'rebound', 'cent', 'growth', 'import', 'subdue'],
```

Fig 9. Lemmatized data output from the dataset

VI. STEPS INVOLVED IN MODELS IMPLEMENTATION AND RESULTS

A. Steps followed for Latent Semantic Analysis (LSI) on dataset

Latent semantic analysis is a method of bringing the latent relationships in the document collection. It works under the assumption that words that have close meaning will appear in the same areas in the document. Then it is sent through a process of Singular value decomposition to get the final list of topics [6].

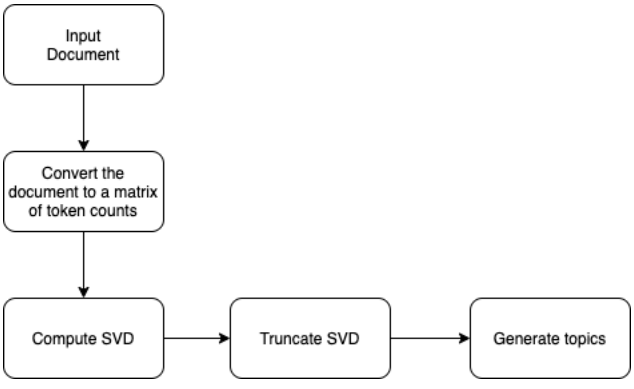


Fig 10. LSI component architecture [16][20]

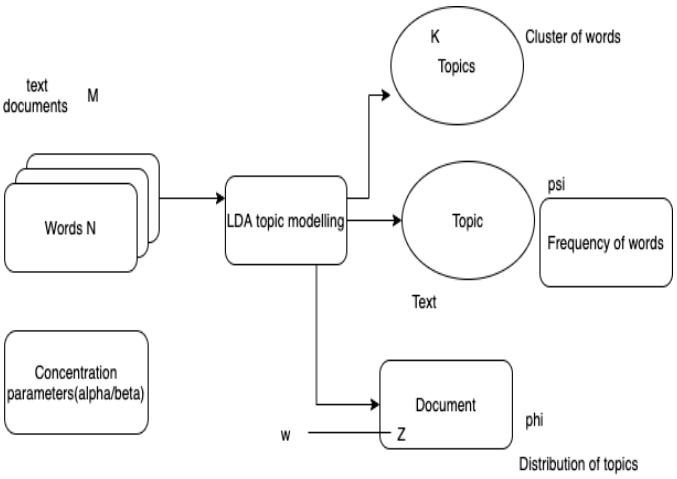
The document is first converted to a matrix of tokens which is then sent to singular value decomposition step for generating the topics.

```
LSI Topic Model output:
Topic 0:
[('samsung', 0.5965427256845779), ('galaxy', 0.43718135519470624), ('new', 0.35773497047887737), (
4), ('android', 0.1131445965789005), ('tab', 0.09503238800600206), ('specs', 0.0822020830451229 4),
2)]
Topic 1:
[('samsung', 0.40518675539276383), ('galaxy', 0.3360689981788052), ('tab', 0.06597366572454924), (
('htc', 0.03511684686370001), ('comparison', 0.028433499001634964), ('mini', 0.021919217032137 23),
092)]
Topic 2:
[('new', 0.7499456105750703), ('facebook', 0.09026603102646773), ('microsoft', 0.0878859941142183)
8), ('health', 0.04631356826750144), ('app', 0.040795311491053275), ('twitter', 0.0344870625830698
02394)]
Topic 3:
[('apple', 0.8674263727289129), ('iphone', 0.15479320266487517), ('ipad', 0.07455839248338599), ('
('stock', 0.053194019070822975), ('patent', 0.05215394358797052), ('deal', 0.04954291514329818), (
```

Fig 11. LSI topics output of dataset

B. Steps followed for Latent Dirichlet Allocation (LDA) on dataset

Latent Dirichlet Allocation is a probabilistic model that takes the document and generates a set of word probabilities. Fig 10 is the architecture of the LDA model processing, and the Hypermeters used for tuning the result [5] [12][13]



K is the number of topics
Psi – frequency of words
Phi – Distribution of topics
M – number of documents to analyse
N – number of words
Alpha – Dirichlet prior concentration parameter of the per document topic distribution
Beta – topic per word distribution parameter
W(i,j) – jth word in the ith document

Fig 12. LDA component architecture diagram [5][16]

Hyper Parameters of LDA

Alpha parameter – documents with higher alpha parameter are considered to have more topics in it. Default value is 1.0

Beta parameter – assumed to have the words in a specific distribution. Default value is 1.0

We haven't used any hyperparameters on this experiment as the model yielded expected results. After the data is cleaned up and preprocessing is done, data will have to transformed to fit the LDA model input format. LDA model requires two primary inputs which is the dictionary id2word and the corpus. The lemmatized output is transformed into a corpus of dictionary. The input is passed to the id2word.doc2bow model which takes the lemmatized output and transforms it to a corpus that contains the of integer ids for the word. Fig 10 shows the corpus of the integer ids created from the dataset. The corpus is the term document frequency of the dataset.

```
[[(0, 1), (1, 1)],
 [(2, 1), (3, 1), (4, 1), (5, 1), (6, 1)],
 [(7, 1), (8, 1), (9, 1), (10, 1)],
 [(11, 1), (12, 1), (13, 1), (14, 1), (15, 1), (16, 1)],
 [(17, 1), (18, 1), (19, 1), (20, 1), (21, 1), (22, 1)],
 [(23, 1), (24, 1), (25, 1)],
 [(26, 1)],
 [(27, 1), (28, 1), (29, 1), (30, 1)],
 [(27, 1), (31, 1), (32, 1)],
 [(33, 1), (34, 1), (35, 1), (36, 1), (37, 1), (38, 1)],
 [(39, 1), (40, 1), (41, 1), (42, 1)],
 [(43, 1), (44, 1), (45, 1)],
 [(46, 1), (47, 1), (48, 1)],
 [(12, 1), (49, 1)],
```

Fig 13. Created corpus of the dataset

Following is the readable form of the corpus

```
[[('affair', 1), ('head', 1), ('lead', 1), ('proctor', 1), ('veteran', 1)],
 [('inversion', 1), ('stop', 1), ('tax', 1)],
```

The corpus and the id2word is further passed to the LDA model which then takes the corpus and converts it to a topic model. Chunksize is the documents size for training chunk. Update_every is the frequency of updating the model parameters. Passes is the number of training passes.

```
[(0,
  '0.042*"cancer" + 0.025*"report" + 0.017*"death" + 0.016*"brain" + '
  '0.013*"eat" + 0.012*"official" + 0.010*"link" + 0.010*"increase" + '
  '0.009*"mental" + 0.008*"smoking"'),
 (1,
  '0.048*"study" + 0.034*"risk" + 0.025*"virus" + 0.018*"drug" + 0.016*"man" + '
  '0.016*"recall" + 0.015*"woman" + 0.015*"disease" + 0.015*"year" + '
  '0.014*"high"'),
 (2,
  '0.032*"health" + 0.031*"case" + 0.024*"heart" + 0.020*"mer" + 0.018*"test" '
  '+ 0.012*"rate" + 0.012*"low" + 0.011*"girl" + 0.011*"skin" + 0.010*"issue"'),
 (3,
  '0.018*"help" + 0.017*"outbreak" + 0.016*"rise" + 0.015*"get" + '
  '0.013*"problem" + 0.013*"need" + 0.010*"fight" + 0.009*"court" + '
  '0.009*"kid" + 0.008*"likely"'),
 (4,
  '0.046*"new" + 0.039*"say" + 0.015*"cigarette" + 0.014*"life" + '
  '0.012*"researcher" + 0.010*"food" + 0.010*"treatment" + 0.010*"day" + '
```

Fig 14. LDA model output of topics of dataset

	Document_No	Dominant_Topic	Topic_Perc_Contrib	Key
0	0	2.0	0.8880	study, drug, new, reveal, reduce, risk,
1	1	0.0	0.9154	cancer, new, help, health, hospital, case

Fig 15. Formatted output of the model keywords

C. Steps followed for Hierarchical Dirichlet model (HDP) on dataset

It is an unsupervised non-parametric and more advanced version of the LDA model that cluster the ungrouped data. Unlike LDA model, it doesn't require the need of specifying the number of topics in the document as it understands from the text itself. For the HDP model, we need the corpus and the dictionary as inputs to train the model and form the topic model. Then we can form the print topics method to see the topics identified on the model. Fig 13 shows the topics identified through the HDP model. The representation in the output can be depicted as topics and its contribution score. The advantage of this model it is more advanced than LDA model and it infers the topics through common base distribution which presents more refined and coherent possible topics from the document. [14]

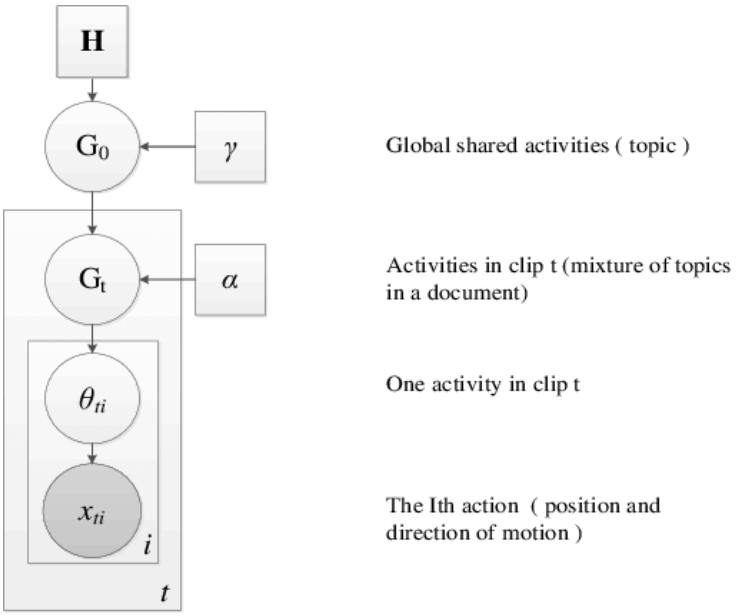


Fig 16. HDP component architecture [17]

```
[(0,
  '0.002*interface + 0.002*mixed + 0.002*noble + 0.002*remain + 0.002*cleanup '
  '+ 0.002*white + 0.002*hall + 0.002*resolution + 0.002*aud + '
  '0.002*inspiration'),
 (1,
  '0.003*outdo + 0.002*shock + 0.002*encrypted + 0.002*take + 0.002*news + '
  '0.002*quarterly + 0.002*cottage + 0.002*graceful + 0.002*train + '
  '0.002*yhoo'),
 (2,
  '0.003*berkshire + 0.003*history + 0.002*infective + 0.002*sharpness + '
  '0.002*goggle + 0.002*chocolate + 0.002*loyal + 0.002*japan + 0.002*await + '
  '0.002*encrypt'),
 (3,
  '0.002*pinpoint + 0.002*system + 0.002*comixology + 0.002*drought + '
  '0.002*helping + 0.002*giant + 0.002*cease + 0.002*airbag + 0.002*pill + '
  '0.002*simplicity'),
 (4,
  '0.002*operation + 0.002*porn + 0.002*malaysian + 0.002*undetermined + '
  '0.002*uninsured + 0.002*riken + 0.002*city + 0.002*sprout + 0.002*mer + '
  '0.002*snoop'),
 (5,
  '0.003*germ + 0.002*revised + 0.002*decide + 0.002*filipino + 0.002*dropbox '
  '+ 0.002*prompt + 0.002*turnaround + 0.002*mass + 0.002*upward + '
  '0.002*regulatory'),
 (6,
  '0.002*tesla + 0.002*doll + 0.002*suspected + 0.002*grade + 0.002*benefit + '
  '0.002*convince + 0.002*barnett + 0.002*event + 0.002*financing + '
  '0.002*stud'),
```

Fig 17. HDP model output of topics from dataset

VII. VISUALIZATION OF MODELS

A. Latent Semantic Analysis visualization

The visualization of Latent Semantic analysis is done by taking the 100 components from the truncated SVD model and they are transformed into 2D words. Which is then plotted using plot library which form the following scatter plot. This is plotted using bokeh[25]

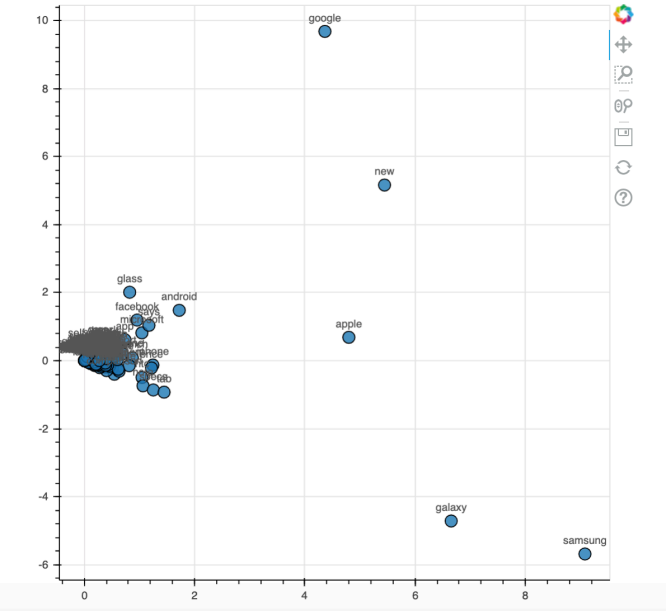
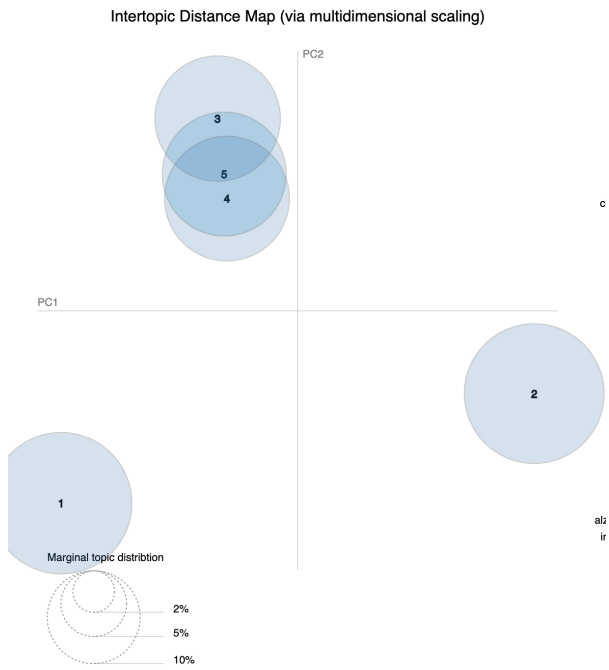


Fig 18. scatter plot of LSI topics

The results show that the topics that are closer to each other and far away from each other. As we could see google, new, apple, galaxy is far from each while facebook, android, Microsoft topics are closer to each other

B. LDA model visualization

The results of LDA model visualized using the library pyLDAvis [21] which takes the 5 topics from the LDA model as input and visualizes the topics is shown in Fig 14



Each of these distributed topics contribute to some portion of the following bar chart of the salient terms.

Fig 19. multidimensional scaling of LDA model output

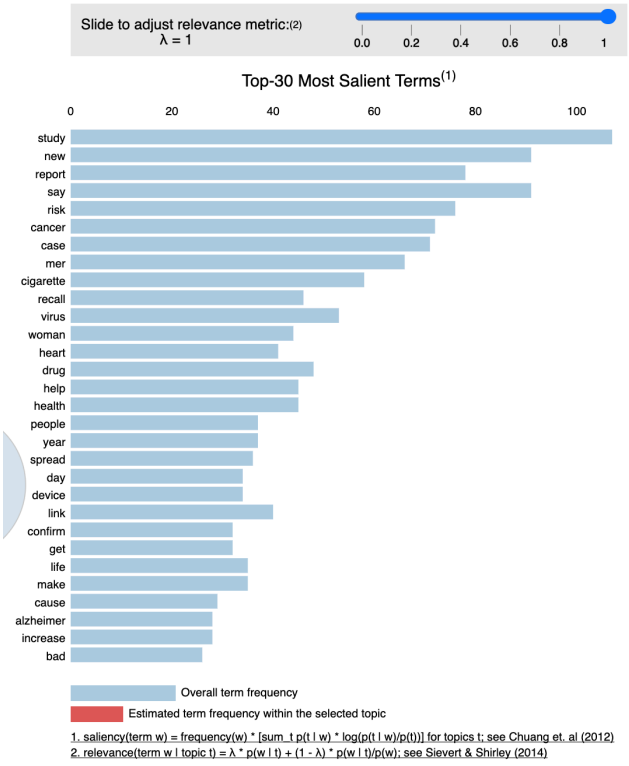


Fig 20. Most salient terms of LDA model output

From the output, it is quite clear that study is the most salient term followed by new, say etc. which all contributes to the final output. This output is plotted using tsne[24] and bokeh[25]

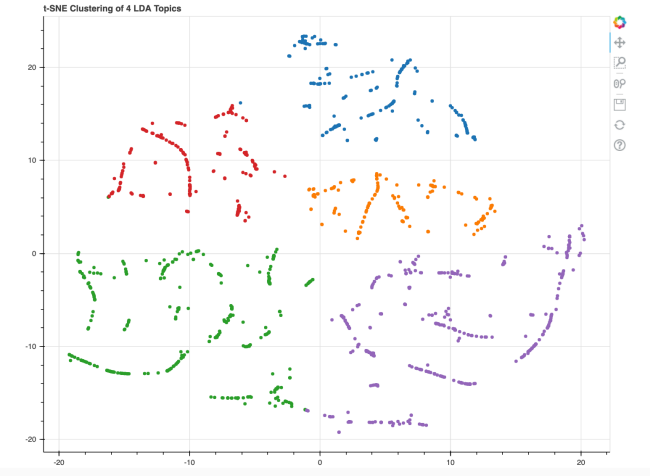


Fig 21. t_sne clustering of 4 topics [15]

C. HDP visualization

As for the HDP visualization there are no proper libraries that presents a more informative view so have shown the formatted list of topics generated from the HDP model.

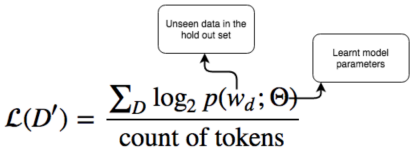
	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords
0	0	15.0	0.5469	caution, quick, spread, instead, sanofi, prepp...
1	1	84.0	0.8896	fingerprint, understand, call, advertiser, inv...
2	2	22.0	0.8345	walled, dairy, wary, saucer, steel, colondar, ...
3	3	136.0	0.8580	window, normal, true, theregister, vitamin, br...
4	4	11.0	0.5033	passenger, swift, presidential, larger, say, g...

Fig 22. Formatted HDP output topics

VIII. EVALUATION MEASURES AND COMPARISON OF MODELS

A. Perplexity

Perplexity is one of the most commonly used topic model evaluation measure. It measures how the model behaves when it has seen a new data set which it has never seen before and can be represented as the normalized log likelihood of the held-out test data. Even optimizing this, wouldn't generate topic accuracy on par with the human interpretation. It can be mathematically expressed as follows [12] [19]



$$perplexity(D') = 2^{-\mathcal{L}(D')}$$

Better model is identified in terms of lower perplexity score.

B. Topic Coherence

In literal terms, a topic is coherent if facts interpreted represents meaningful information. Likewise, Topic coherence measures the similarity of the high scoring words semantically. This measure enables us to differentiate the human perceptible topics to the topics that were infered statistically by the model. For computing this, we have to follow the following steps.

1. choose the most repeating words in a every topic
2. Determine pairwise scores UCI for the selected words as well as the overall sum of all the pairwise scores to compute the coherence of the topic
3. The final step to determine the coherence score is to take a mean of all the topics found in the model. It can be mathematically expressed as [12]

$$\text{Coherence} = \sum_{i<j} \text{score}(w_i, w_j)$$

A topic will have a good coherence score if it has sufficient number of topics and higher the topic scores the more interpretable the topic distribution is. Following are the perplexity scores for the document. From the scores HDP model has the highest score of close to 0.77 and LDA model has the second-best performing coherence score of 0.60 and LSI model is the least performer with 0.40 respectively

```
Perplexity of LDA model:  -8.67815142398039

Coherence Score of LDA model:  0.6013371529103662

Coherence Score of HDP model:  0.7728857435875127

Coherence Score of LSI model: 0.4090677901903302
```

Fig 18. Coherence scores of the models on the dataset.

Following are the coherence plots and the final coherence comparison plotted using mathplotlib library [23]

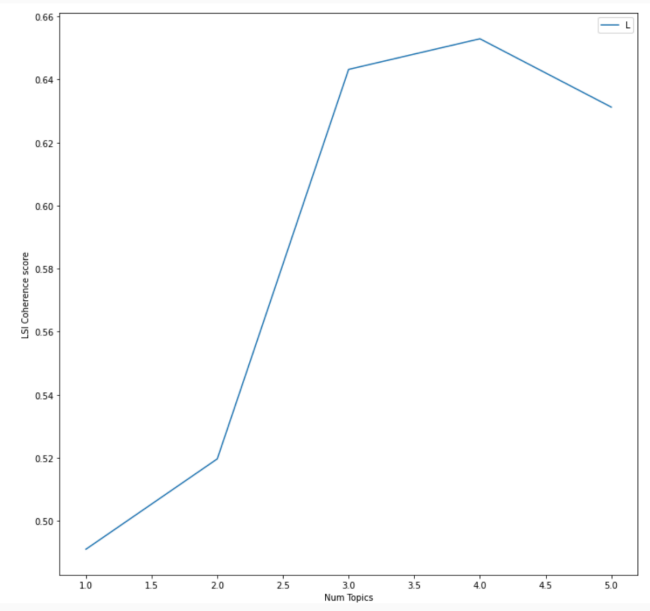


Fig 23. LSI model coherence score plot

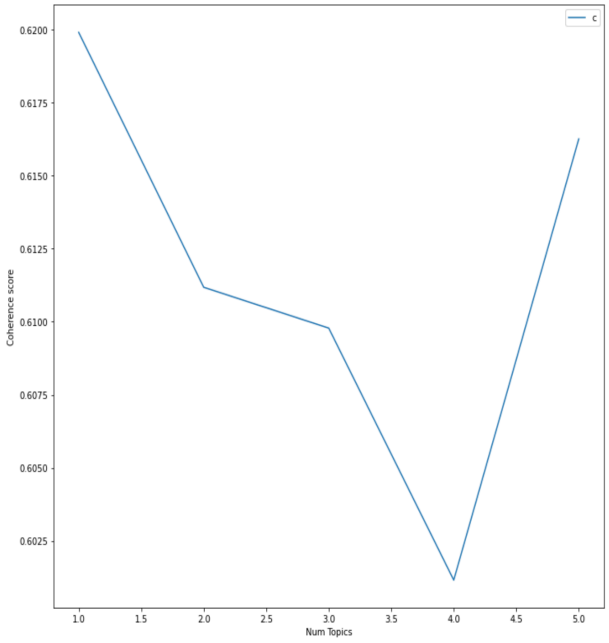


Fig 24. LDA model coherence score plot

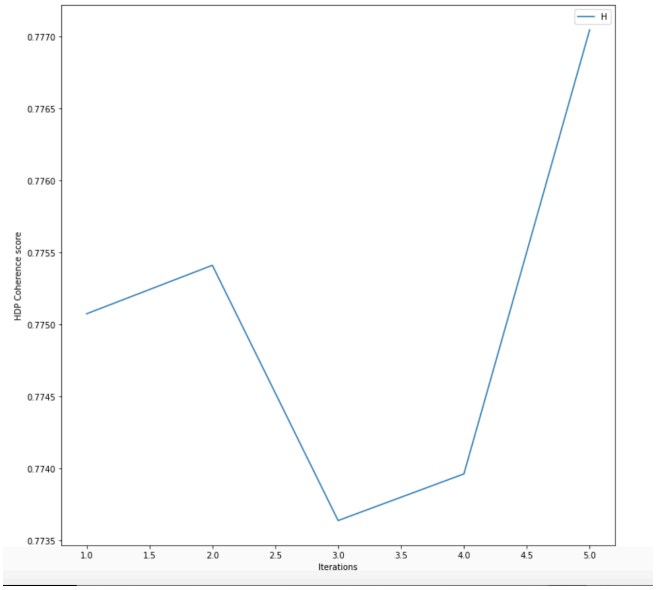


Fig 25. HDP model coherence score plot

For HDP model since we don't have the to specify the number of topics. So, we have iterated the HDP model to see the different coherence values computed from it.

C. Comparison results and Discussion

Following are the overall coherence value comparisons for the models

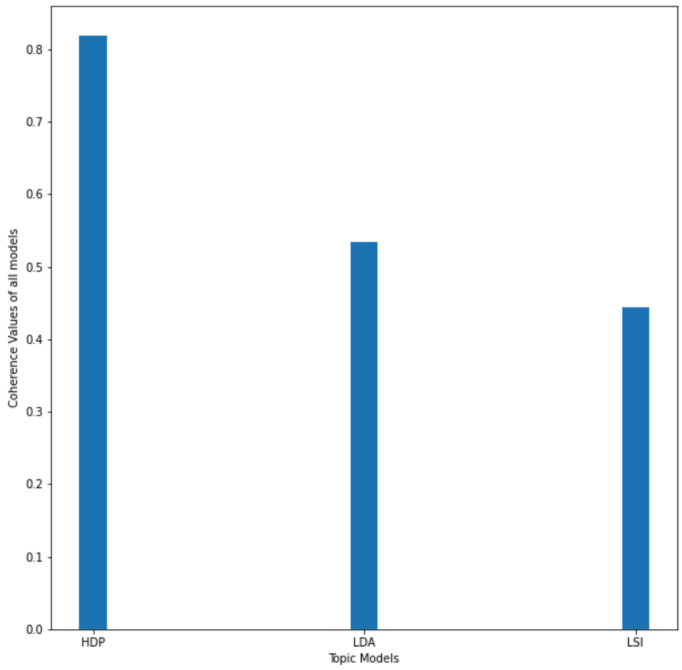


Fig 25. plot showing coherence values comparison of all three evaluated models

From the three compared models it is evident from the plot that the coherence value of HDP is higher which tell us that the HDP model is a better model for generating topic models. LDA and LSI models has coherence values of little over 0.6 and 0.4 to 0.5 for LSI model. This comparison also proves that the HDP model is a better coherent model and can produce efficient results than the other two models in context perspective.

IX. Jupyter notebook parameters

A. Mandatory parameters

Input text file name: Necessary to get the text for topic modeling.

Assumptions: Need to include them in the specific format as attached in the excel. Code template reference [4]

X. Conclusion

From the above results of the comparison, we can draw a conclusion that HDP is a much better model when it comes to coherence score. However, if we have to specify the number of topics then LDA is would be the preferred model as it has better coherent score than LSI model. Choosing the right model depends requirements of the user.

XI. Future work

Each model has its own advantages and limitations. For further model evaluations some more metrics needs to be evaluated for all the models along with coherence which could help us identify the appropriate model. Along with this the visualizations of HDP model also has to be improved.

XII. References

[1] Analytics Vidhya, "Topic Modelling In Python Using Latent Semantic Analysis," *Analytics Vidhya*, Oct-2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/. [Accessed: 15-Dec-2020]

[2] "Introduction to Topic Modeling," *MonkeyLearn Blog*, 26-Sep-2019. [Online]. Available: https://monkeylearn.com/blog/introduction-to-topic-modeling/. [Accessed: 15-Dec-2020]

[3] "Gensim - Quick Guide - Tutorialspoint," *Tutorialspoint.com*, 2019. [Online]. Available: https://www.tutorialspoint.com/gensim/gensim_quick_guide.htm. [Accessed: 15-Dec-2020]

[4] akashram, "Topic Modeling - Intro & Implementation," *Kaggle.com*, 07-Nov-2019. [Online]. Available: https://www.kaggle.com/akashram/topic-modeling-intro-implementation/notebook. [Accessed: 15-Dec-2020]

[5] Shashank Kapadia, "Topic Modeling in Python: Latent Dirichlet Allocation (LDA)," *Medium*, 15-Apr-2019. [Online]. Available: https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0. [Accessed: 15-Dec-2020]

[6] H. Wallach, "topic modeling" [Online]. Available: https://people.cs.umass.edu/~wallach/talks/topic_modeling.pdf

[7] pandas-profiling, "pandas-profiling/pandas-profiling," *GitHub*, 30-Nov-2020. [Online]. Available: https://github.com/pandas-profiling/pandas-profiling. [Accessed: 16-Dec-2020]

[8] https://www.facebook.com/rtipaday, "Topic Modeling in Python with Gensim," ML+, 26-Mar-2018. [Online]. Available: https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/. [Accessed: 15-Dec-2020]

[9] "gensim," *PyPI*, 04-May-2020. [Online]. Available: https://pypi.org/project/gensim/. [Accessed: 16-Dec-2020]

[10] "nltk.lm package — NLTK 3.5 documentation," *Nltk.org*, 2020. [Online]. Available: https://www.nltk.org/api/nltk.lm.html. [Accessed: 15-Dec-2020]

[11] "Plotly Python Graphing Library," *Plotly.com*, 2020. [Online]. Available: https://plotly.com/python/. [Accessed: 16-Dec-2020]

[12] Wuraola Oyewusi, "Exploring Topic Modelling with Gensim on the Essential Science Indicators Journals List.," *Medium*, 25-Jan-2019. [Online].

Available: https://medium.com/@oyewusiwuraola/exploring-topic-modelling-with-gensim-on-the-essential-science-indicators-journals-list-1dc4d9f96d9c. [Accessed: 15-Dec-2020]

[13] Sooraj Subrahmannian, "Learn to Find Topics in a Text Corpus - Sooraj Subrahmannian - Medium," Medium, 16-Apr-2018. [Online]. Available: https://medium.com/@soorajsubrahmannian/extracting-hidden-topics-in-a-corpus-55b2214fc17d. [Accessed: 15-Dec-2020]

[13] "Gensim: topic modelling for humans," *Radimrehurek.com*, 2010. [Online]. Available: https://radimrehurek.com/gensim/models/ldamodel.html. [Accessed: 15-Dec-2020]

[14] "Gensim: topic modelling for humans," *Radimrehurek.com*, 2011. [Online]. Available: https://radimrehurek.com/gensim/models/hdpmodel.html. [Accessed: 15-Dec-2020]

[15] https://www.facebook.com/rtipaday, "Topic modeling visualization - How to present results of LDA model? | ML+," *ML+*, 04-Dec-2018. [Online]. Available: https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/#13.-t-SNE-Clustering-Chart. [Accessed: 15-Dec-2020]

[16] diagrams.net - free flowchart maker and diagrams online, "Flowchart Maker & Online Diagram Software," *Diagrams.net*, 2020. [Online]. Available: https://app.diagrams.net/#G1Pc9i5ITsZn4-ClKD5N13DhV3m8RXP6xk. [Accessed: 15-Dec-2020]

[17] unknown, "Fig. 2: A graphical representation of HDP model. It consists of two...," *ResearchGate*, 09-Feb-2018. [Online]. Available: https://www.researchgate.net/figure/A-graphical-representation-of-HDP-model-It-consists-of-two-Dirichlet-Processes-The_fig2_323118560. [Accessed: 15-Dec-2020]

[18] "Submission History - CSCI6515 - Machine learning for Big Data (Sec 1) - 2020 Fall - Dalhousie University," *Brightspace.com*, 2020. [Online]. Available: https://dal.brightspace.com/d2l/lms/dropbox/user/folders_history.d2l?db=86706&grpid=160044&isprv=0&bp=0&ou=131887. [Accessed: 16-Dec-2020]

[19] Shashank Kapadia, "Evaluate Topic Models: Latent Dirichlet Allocation (LDA)," *Medium*, 19-Aug-2019. [Online]. Available: https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0. [Accessed: 16-Dec-2020]

[20] G. A. León-Paredes, L. I. Barbosa-Santillán, and J. J. Sánchez-Escobar, "A Heterogeneous System Based on Latent Semantic Analysis Using GPU and Multi-CPU," *Scientific Programming*, vol. 2017, pp. 1–19, 2017, doi: 10.1155/2017/8131390. [Online]. Available: https://www.hindawi.com/journals/sp/2017/8131390/. [Accessed: 16-Dec-2020]

[21] "pyLDAvis," *PyPI*, 05-Jun-2018. [Online]. Available: https://pypi.org/project/pyLDAvis/. [Accessed: 16-Dec-2020]

[22] J. Xu, "Topic Modeling with LSA, PLSA, LDA & lda2Vec - NanoNets - Medium," *Medium*, 25-May-2018. [Online]. Available: https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05. [Accessed: 16-Dec-2020]

[23] "Matplotlib: Python plotting — Matplotlib 3.3.3 documentation," *Matplotlib.org*, 2012. [Online]. Available: https://matplotlib.org/. [Accessed: 16-Dec-2020]

[24] "sklearn.manifold.TSNE — scikit-learn 0.23.2 documentation," *Scikit-learn.org*, 2014. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html. [Accessed: 16-Dec-2020]

[25] Bokeh Contributors, "Bokeh 2.2.3 Documentation," *Bokeh.org*, 2019. [Online]. Available: https://docs.bokeh.org/en/latest/index.html. [Accessed: 16-Dec-2020]