Embedding for Evaluation of Topic Modeling - Unsupervised Algorithms

[1]Ms.Ananya Srivastava, [2]Ms.Lavanya Gunasekar, [3]Mrs.Bagyalakshmi.V

[1]Data Science Developer, [2]Data Scientist, [3]Principal Scientist
Tata Consultancy Services Pvt. Ltd, Chennai, India

***Abstract*: Topic Modeling is one of the most popular techniques used for text mining in Natural Language Processing. Topic modeling refers to the task of identifying topics that best describes a set of documents. It will classify data based on a particular topic and determine the relationship between tokens. This is done by extracting the patterns of word clusters and frequencies of words in the document. It has enjoyed success in various applications in machine learning, natural language processing (NLP), and data mining for almost two decades. There are several algorithms for implementing topic modeling. Most common techniques are LDA – Latent Dirichlet Allocation, LSA or LSI – Latent Semantic Analysis or Latent Semantic Indexing. In this paper, we have proposed the Word Embedding Topic Evaluation methodology which will help in identifying the efficient outcomes with better accuracy. It outperforms existing document models that are generally used in measuring topic evaluation such as coherence score, perplexity etc., in terms of topic quality and predictive performance.**

***Keywords*: Evaluation of Topic Modeling, Word Embedding, Word2vec.**

I. INTRODUCTION

Natural Language Processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, how to program computers to process and analyze large amounts of natural language data. When it comes to Topic Modeling, it provides us with methods to organize, understand and summarize large collections of textual information. It helps in discovering hidden topical patterns that are present across the collection. Some of the topics makes sense, some of them do not. These models can learn subjects that are highly interpretable, semantically coherent and can be used similarly to subject headings. But sometimes learned topics are subset of words that do not convey much useful information.

Algorithms and Techniques used in Improving Topic Modeling:

Some algorithms used for Topic Modeling tasks are Latent Dirichlet Allocation, Latent Semantic Analysis, Correlated Topic Modeling, and Probabilistic Latent Semantic Analysis. But when it comes to evaluation the Coherence score, Perplexity are the primary evaluation measures for the results.

Coherence score is defined as how many times two words appear together in documents, and how many time word appeared alone. The greater the number, the better is coherence score. On the other hand, perplexity is one of the intrinsic evaluation metrics in which the model is used to assign a likelihood score to unseen data, and this score is measured as normalized log-likelihood of a held-out test data.

However, recent studies have shown that predictive likelihood and human judgment are often not correlated. These metrics are not sufficient and good indicator for assessing the quality of topics. So, we need a generalized standard system of measurement to estimate the topic terms with respect to any API outcomes.

The most common way to figure out a probabilistic model is to measure the log-likelihood of a held-out test set which will be mostly utilized for LDA models. In this paper, there is a unique way of measuring and evaluating the outcomes of all the Topic Modeling methodology which has outperforms the existing standards.

II. RELATED WORK

This section provides overview on the existing assessing measure for Topic Modeling and the proposed method:

- Coherence Score
- Perplexity Score
- Word Embedding Evaluation Measure

In recent years, for a set of words (or topics), different word similarities have been used in coherence formulas for identifying best tokens from this collection of words and perplexity score to compare models.

Coherence Score

It gives the measure of how often words support each other in the document or how often the two words appear in a corpus. Topics are not assured to be well interpretable, therefore, coherence measures have been proposed to distinguish between good and bad topics. Studies of topic coherence so far are limited to measure the score for pairs of individual words. The fact is not considered that how well derived topics (i.e., the top words) can be interpreted by humans.

Topic models have been combined with coherence measures by introducing specific priors on topic distributions. However, a coherence measure based on word pairs would assign a good score. Existing methods calculate the score based on the semantic similarity of topic-related words.

One of the most popular coherence metrics is called CV. It creates content vectors of words using their co-occurrences and, after that, calculates the score using normalized pointwise mutual information (NPMI) and the cosine similarity. This metric is popular because it is the default system in the Genism topic coherence pipeline module.

There is no way to determine whether the coherence score is good or bad. The score and its value depend on the data that it is manipulated from. For instance, in one case, the score of 0.5 might be good enough to judge but in another case it is not. The only rule is that we want to maximize this score.

Perplexity Score

In Natural Language Processing, it is applied to measure or evaluate how well topic models are predicting. It is one of the intrinsic validation metrics and is widely used for language model evaluation that captures how surprised a model is of new data it has not seen before.

Focusing on the log-likelihood part, you can think of the perplexity metric as measuring how probable some new unseen data is given the model that was learned earlier.

- Topic Coherence measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference. If the number of topics increases the coherence score increases but we could have repeated keywords in the topic.

- Topic Coherence is the intrinsic measure of 'genism' API which can't be used for other outcomes generated with different APIs such as sklearn, nltk, spaCy etc.,

- Recent studies have shown that predictive likelihood (or equivalently, perplexity) and human judgment are often not correlated, and even sometimes slightly anti-correlated. Optimizing for perplexity may not yield human interpretable topics.

The limitations of these metrics likely to provoke a universal measure for all algorithms in topic modeling.

To overcome the existing challenges, we proposed a novel method to measure topic evaluation using Word Embedding: Firstly, topic model output has been generated using any API and with any algorithms like LDA, LSI, SVD, NMF etc. Then, similarity measure was calculated for each topics by pairing the tokens in each topic and calculating the cosine similarity measure between them. After this, average measure topic score has been calculated by adding up all topic scores divided by number of topics(k) in result.

Irrespective of any APIs (NLTK/sklearn/genism) or algorithms (LDA/NMF/LSI etc.,), we could analyze the affirmation of topics generated using this technique. From the average topic score, we could get an idea about how efficient the topics been built. If the similarity score is high, then the tokens are closely related and well connected with respect to context. Completely automated where human effort is eliminated to find the best topics out of different topic modeling results.

III. METHODOLOGY

In this section, we will be discussing about the different steps involved to feed the data to the various Deep Learning and Machine Learning models for generating topics across the documents. The dataset that has been used here belongs to Ecommerce - Fashion industry.

The steps involved are:

- Data Collection
- Data Preprocessing
- Feature Extraction
- Model Implementation
- Evaluation Metric for Topic Modeling

Data Collection

The dataset holds 4429 rows and 456 categories. This includes 1504 negative comments and 2925 positive comments. The other attributes were Product Name, Product Id, Ratings etc. The question we were interested in solving is how often these words come up in a single document. This will allow us to categorize each document to a particular topic or a theme.

Data Preprocessing

In this section, all data points were text data that possess the English Language linguistic properties. The pre-processing step was applied to clean up the data and then convert the extracted information into a structured format to analyze the patterns (visible and hidden) within the data. The pre-processing steps are supported in Stanford's NLTK Library and contain the following patterns:

➢ Tokenize: Segmented the data by converting the sentence to tokens by replacing the punctuations with appropriate strings.

➢ Stop word and lemmatization elimination: (1) Removed stop words, (2) Filtered the Noun, Adjective, Adverb, Verb POS tags from tokens and (3) Lemmatize the extracted tokens to root word.

➢ Also, all the contractions present in the reviews were expanded (For example "I'll" with "I will", "wasn't" with "was not", etc.) In addition, in this phase the entire data frame content was converted to lowercase for better data processing.

Feature Extraction

Now, the cleaned data points were used to perform the feature extraction and it was done by generating the TF-IDF method.

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. TF*IDF do not convert raw data directly into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector.

TF = Number of occurrences of a word (i) in document(j) / Total words in document (j)

IDF = log (Total number of documents / number of documents containing the word (i))

Model Implementation

Here, various topic modeling algorithms like LDA, LSA, NMF using genism, sklearn python libraries were executed. The topic modeling outputs for different number of topics(k) - values ranging from 4 to 13 are generated.

➢ LDA: Latent Dirichlet Allocation (LDA) is a popular topic modeling technique to extract topics from a given corpus. The term latent conveys something that exists but is not yet developed. In other words, latent means hidden or concealed.

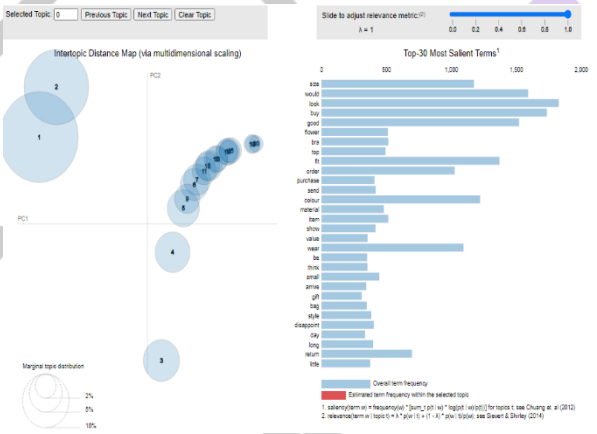


Fig. 1 LDA Output

Here in the fig 2. PyLDA library has been imported that provides a dynamic representation of the topics and the word frequency. Circles represents the topic and the search box allow to select any topic to view the words and their respective frequency in the left bar chart.

➢ NMF: The NMF technique examines documents and discovers topics in a mathematical framework through probability distributions. As an illustration, we first have matrix X as our data. This matrix is represented by two smaller matrices W and H, which, when multiplied, approximately reconstruct X. We implement and run the NMF algorithm on this data by minimizing the following divergence penalty, where W and H contain nonnegative values:

$$D(X||WH) = -\sum_{i}\sum_{j}[X_{ij}In(WH)_{ij} - (WH)_{IJ}]$$

After running the algorithm, we want to normalize the columns of 'W' so they sum to one. This is also to ensure that we get probabilistic distributions with no values greater than zero. Now, say we pick 10 words for each topic. For each column of W,

we list the 10 words having the largest weight and show the weight to show our expected probabilistic distributions. The i^{th} row of W corresponds to the i^{th} word in the "dictionary" provided with the data.

➢ LSA: Latent Semantic Analysis, or LSA, is one of the foundational techniques in topic modeling. The core idea is to take a matrix of what we have documents and terms and decompose it into a separate document-topic matrix and a topic-term matrix. Consequently, LSA model replace 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'. LSA is quick and efficient to use. But it has a few primary drawback that it works good on large dataset and less efficient representation of embeddings.

➢ SVD: We would clearly expect that the words that appear most frequently in one topic would appear less frequently in the other — otherwise that word wouldn't make a good choice to separate out the two topics. Therefore, we expect the topics to be orthogonal. The SVD algorithm factorizes a matrix into one matrix with orthogonal columns and one with orthogonal rows (along with a diagonal matrix, which contains the relative importance of each factor). SVD is an exact decomposition, since the matrices it creates are big enough to fully cover the original matrix.

The frequently used algorithm in Topic Modeling is LDA that digs out topic probabilities from statistical data available. While using these methodologies, there are some challenges. The major challenge is the developer should know the fixed number of topics/themes which is not possible in real world. Hence, approaches such as the LDA or LSA require conditioning to handle issues like optimization of number of topics(k), overfitting, non-linearity, and discovery of too many generic words.

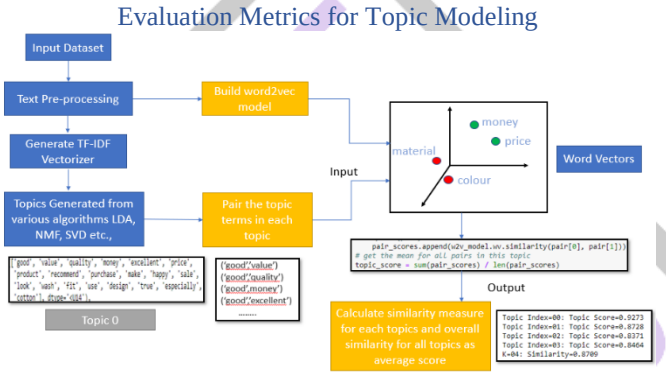


Fig. 2 Architecture for Topic Modeling Evaluation

Existing techniques shows the generalization power of model on unseen data. There are several ways to evaluate topic models, including:

➢ Human judgment
➢ Observation-based, e.g. observing the top 'N' words in a topic.
➢ Interpretation-based, e.g. 'word intrusion' and 'topic intrusion' to identify the words or topics that belongs to a theme.
➢ Quantitative metrics – Perplexity (held out likelihood) and coherence calculations.
➢ Mixed approaches – Combinations of judgment-based and quantitative approaches.

Coherence score and perplexity are the prevailing evaluation standard for the result analysis. Due to the limitations of these metrics, we are suggesting an easy and effective method to evaluate the topic modeling results.

Limitations of Coherence score:

➢ Comparability – The coherence pipeline allows the user to select different methods for each part of the pipeline. This, combined with the unknown variability of coherence scores, makes it difficult to meaningfully compare different scores, or scores between different models.

➢ Reference corpus – The choice of reference corpus is important. In cases where the probability estimates are based on the reference corpus, then a smaller or domain-specific corpus can produce misleading results when applied to a set of documents that are quite different to the reference corpus.

Word2vec is one of the most famous approaches for word embedding using shallow neural network. It can be obtained using two methods, one is Common Bag of Words (CBOW) and other one is Skip Gram. We have implemented Common Bag of Words for identifying the association between each word in different topics. The higher the correlation between words with respect to context increases the scoring.

Word Embedding for proximity measure: The steps involved are list below

➢ Get the complete feature names from TF-IDF matrix
➢ Calculate actual weight of the features

i. tfidf.sum(axis=0) – This will sum up the values across the rows and produce the overall weightage against the term.
ii. Now weightage for each term would be generated.
iii. Sort the terms based on weightage

➢ Generate word vectors using word2vec model from the pre-processed data

Average Similarity Measure: In this study, we have paired the token in each topic and calculated the cosine similarity between the pairs. Next, iteration is done for all pairs in the first topic. We have calculated the topic_0 score by adding up all pair scores divided by number of pairs in topic_0. Similarly, average topic score for each topic is calculated. The final measure is manipulated in which average topic score is calculated by adding up all topic scores divided by number of topics in result. From the average topic score, we could get idea about how efficient the topics are built. If the similarity score is high, then the topics are closely related and well connected.

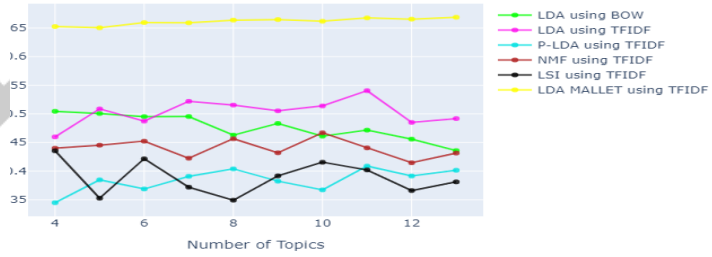
The novelty that has been proposed was described in a mathematical term as follow:

➢ w2v_model= word2vec model output for input data set after removing the noises.
➢ Pair score: Pairs are manipulated in each topic for calculating the association.
➢ Topic score = sum(pair_scores) / length(pair_scores)
➢ Similarity score = sum(topic_score) / Length (topics in topic modeling). This will provide the similarity measure for the generated set of words.

Furthermore, word similarity can be used to test model's robustness against lexical ambiguity, as a dataset aimed at testing multiple senses of a word can be created. Therefore, word2vec can capture the contextual value between words from the training of a large corpus.

IV. RESULT AND DISCUSSION

One of the shortcomings of perplexity is that it does not capture context, i.e. it does not capture the relationship between words in a subset of words or topics in a document. Hence the resultant technique scoring – average proximity score is compared against coherence score.



. **Fig. 3** Coherence Score for Various Models

Figure 3 shows the coherence score for various models where LDA Mallet using TFIDF has outperformed other models and second comes the LDA using TFIDF intersecting LDA using BOW model between number of topics(k) 4 to 6.

df_avg

	API	Algorithm	Avg_Coherence_Score	Avg_Similarity_Score
0	gensim	LDA Mallet using TFIDF	0.66141	0.96424
1	gensim	LDA using BOW	0.47663	0.90201
2	gensim	LDA using TFIDF	0.50299	0.92708
3	gensim	LSI using TFIDF	0.38873	0.85674
4	gensim	NMF using TFIDF	0.44024	0.91074
5	gensim	P-LDA using TFIDF	0.38446	0.88708
6	sklearn	LDA using TFIDF	NaN	0.89316
7	sklearn	NMF using TFIDF	NaN	0.92332
8	sklearn	SVD using TFIDF	NaN	0.87555

Table 1 . Various model analysis: API-Algorithms (genism/sklearn) Vs Average Similarity Score

This table shows the average coherence score and average similarity score of each algorithm and the API used is gensim for topic modeling. For each algorithm, the value of k – number of topics are considered from 4 to 13. Multiple iterations are carried out for optimum score. Then the average score is manipulated for each algorithm.

Even though coherence score is high, topic results are not good. There are lot of repeated words occurring in the topic.

Technology advancement over the existing evaluation techniques:

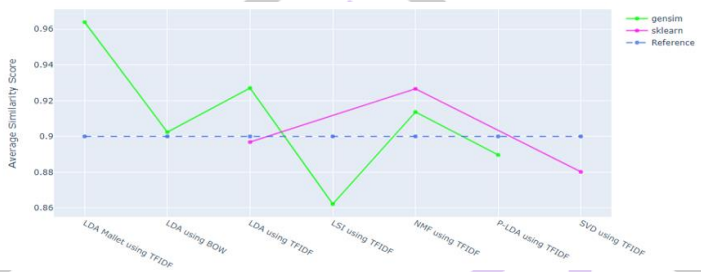
➢ Robustness against lexical ambiguity: As Similarity scores are normalized by the vector length, it is robust to scaling. It is computationally inexpensive Thus, it's easy to compare multiple scores from a model and can be used in word model's prototyping and development. This is needed to avoid meaningless representations from conflicting properties that may arise from the polysemy of words.

➢ Efficiency: Similarity measure is computationally efficient. Most models are created to solve computationally expensive downstream tasks. This evaluation technique is simple yet able to predict the downstream performance of a model

➢ Statistical significance: The performance of word embedding evaluation technique with respect to an evaluator should have enough statistical significance, or enough variance between score distributions, to be differentiated. This is needed in judging whether a model is better than another and helpful in determining performance rankings between models.

➢ Evaluating topic models is unfortunately difficult to do. There are various approaches available, but the best results come from human interpretation. This is a time-consuming and not much reliable. Hence the manual effort is reduced and reliability on performance is improved with this method.

➢ Existing metrics like coherence score, perplexity is statistical metric which has restrictions with limited API and algorithms. This method can extensively be used with any API and algorithm evaluation.



. **Fig. 4**. Various model analysis: API-Algorithms (genism/sklearn) Vs Average Similarity Score

The above graph shows the embedding average similarity measures for topics generated using different models.

➢ LDA mallet is providing highest average similarity measure with ~ 0.98 which looks to be overfitting of data and some terms are repeated between topics
➢ All LDA models are performing good on an average of above 0.9 average similarity measure.
➢ NMF model contributes ~ 0.92 average similarity measure in both APIs – sklearn and genism for our corpus with meaningful topics.

V. CONCLUSION

Fig. 5 .Average Coherence Score and Average Similarity Score for Various models

The Figure 5 shows the difference between average coherence score and average similarity score.

It is clear that the choice of embedding algorithm, model variant, and background corpus has a large impact on the resulting coherence values, which could potentially influence topic model parameter selection choices. And ultimately affect the interpretations made from the topics identified on a given corpus.

Our study has produced better insights compared to existing ones. This could be extended to any model irrespective with any API. The embedding technique can be replaced with state-of-the-art techniques like BERT, GloVe, etc., for the best enhancement of the disrupted method. The best performing methods provide reliable estimates of topic similarity compared with human performance. This could be used in preference to the word probability distribution measures and hybrid approaches.

REFERENCES

[1] Martin Popel, David Mareček "Perplexity of n-Gram and Dependency Language Models", Published by: International Conference on Text, Speech and Dialogue, March 2010.

[2] Noah A. Smith "Adversarial Evaluation for Models of Natural Language", Published by: arXiv.org, 1 Jul 2012.

[3] David Newman, Sarvnaz Karimi and Lawrence Cavedon "External Evaluation of Topic Models" Published by: institute of Electrical and Electronics Engineers, 7-10 Dec. 2013.

[4] Frank Rosner, Alexander Hinneburg, Michael Röder, Martin Nettling "Evaluating topic coherence measures", Published by: Neural Information Processing Systems Foundation (NIPS 2013) - Topic Models Workshop, December 2013.

[5] Muhammad Omar, Byung-Won On, Ingyu Lee, Gyu Sang Choi "LDA topics: Representation and evaluation", Published By: Journal of Information Science, June 2015.

[6] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, Liang Zhao "Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey" Published by: arXiv.org, 6 Dec 2018.

[7] Damir Korencic, Strahil Ristov, Jelena Repar and Jan Snajder "A Topic Coverage Approach to Evaluation of Topic Models" Published by: arXiv.org, Dec 2020.

[8] Gina V. Acosta Gutiérrez "A Comparative Study of NLP and Machine Learning Techniques for Sentiment Analysis and Topic Modeling on Amazon Reviews", Published by: The Journal of Supercomputing, April 2020.

[9] Subhashini Gupta, Grishma Sharma "Topic Modeling in Natural Language Processing", Published by:International Journal of Engineering Research and Technology, June 2021