Topic Modeling with LDA and NMF

Name

Yiyang Bi

Halıcıoğlu Data Science Institute

Supervisor:

Dr. Tsui-Wei Weng

Posted:

Jan 1, 2022

1. Introduction

Topic modeling is an unsupervised machine-learning technique for finding the underlying semantic/latent topics given a large collection of corpus/documents[1]. As a powerful technique for text analysis, topic modeling is one of the most important techniques across various aspects of applications and research(shown in sub-section 1.1).

Latent Semantic Analysis (LSA), is one of the first algorithms that tried to solve topic modeling problems. LSA decomposed the document-term matrix into a separate document-topic matrix and a topic-term matrix by applying Single Value Decomposition[2].

Latent Dirichlet allocation (LDA) is one of the Bayesian probabilistic topic models (BPTMs) for handling topic modeling problems. LDA is generally an improvement on LSA, which estimates topic and topic terms in a probabilistic way. In general, LDA yields better-predicted accuracy than LSA on topic modeling task[3].

In this project, our team is also going to implement Non-Negative Matrix Factorization (NMF), which is a linear algebra approach similar to Latent Semantic Analysis (LSA), but with some improvements upon LSA.

1.1 Applications:

Topic Models have a wide variety of applications:

1. Linguistic science[4]
2. Analyze political attention[5]
3. Recommendation system[6]
4. Adverse Drug Reaction Prediction[7]
5. Discovering newsworthy information[8]

6. Software evolution[9] and source code analysis[10]
7. Sentiment analysis[11]
8. Crime prediction/evaluation[12]

And many more...

---

2. Problem Formulation

2.1 Relation to numerical linear algebra:

The most traditional way to represent text as numbers is by using Term Frequency Inverse Document Frequency(TF-IDF). **TF-IDF** is a product of **term frequency (TF)** and **inverse document frequency (IDF)**. **Term frequency (TF)** is the number of occurrence words/terms in a given corpus. However, the higher frequency of words does not necessarily imply the terms/words are important, in fact, some joint words (e.g. and) is not providing any importance to the topics Thus, we need to apply **inverse document frequency (IDF)**, which is help to regularize the weights of the terms/words that appear too often but cannot represent any meaning for the topics. Another advantage of applying **IDF** is that it will raise the attention of least occurring words which might actually have much more meaning for finding the topic of the context.

The following mathmatical representation of finding the values of **TF-IDF**[14]**:**

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

Which **TF** can be represented by:

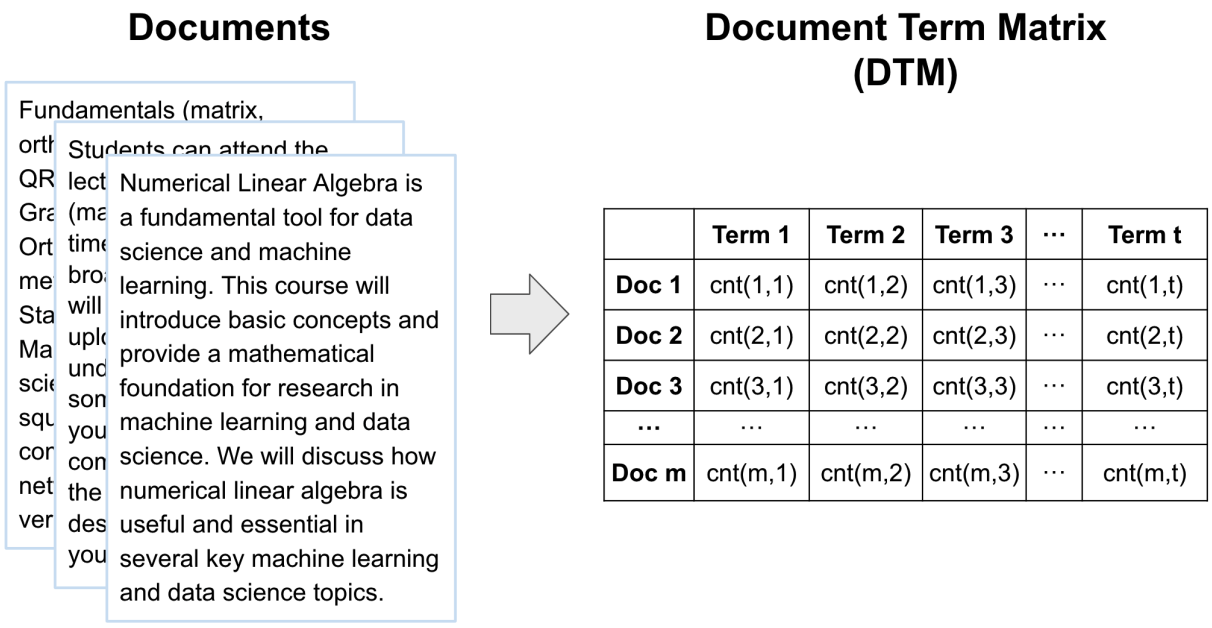
$$tf(t, d) = log(1 + freq(t, d))$$

IDF can be represented as:

$$idf(t, D) = log(\frac{N}{count(d \in D : t \in d)})$$

where N is the number of documents, t is the specific word d is the given document, which is a subset of the whole document set D.

By using the **TF-IDF** transformation, we can convert the texts/documents into a **Document Term Matrix (DTM)**, which will be applied to the decomposition algorithms for finding the latent topics**.**



The above figure shows the transformation from textual documents into **Document Term Matrix (DTM)**. In the table of DTM, each row is a document, and each column is word in the document, and each cell contains the count of the term that appears in the corresponding document. The word count considers the **TF-IDF score** mentioned above.

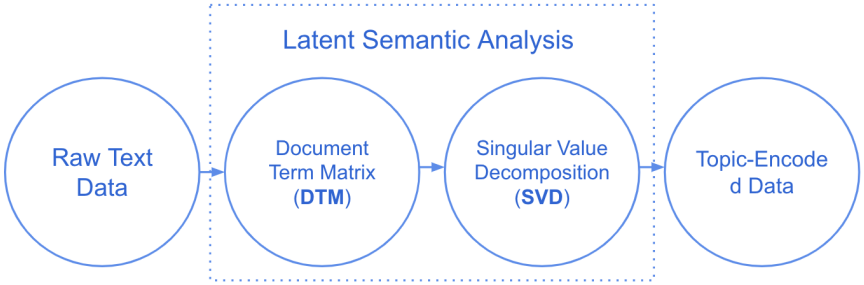
Now we have input $\mathbf{A}_{n\times m}$ the question is: **How are we finding the topics using the input?**

To solve this problem we will apply **Latent Semantic Analysis**, which will be discussed in the next subsection.

2.2 Approach description:

2.2.1 Introduction of Latent Semantic Analysis (LSA)

Latent Semantic Analysis(LSA) is one of the most conventional and well-known approaches for solving textual decomposing problems of topic modeling. It is a mathematical method by using linear algebra for computer modeling and simulation of the meaning of words by analyzing the representative corpora of documents.



The goal of Latent Semantic Analysis(LSA) is to analyze the raw text data and transform it into DTM, and finally extract latent topics. During the decomposing process, a matrix factorization technique called **Single Value**

Decomposition (SVD) can be utilized. SVD is a mathematical approach that is used to reduce the number of rows while preserving the similarity structure among columns[15].

2.2.2 Decomposes approach and process

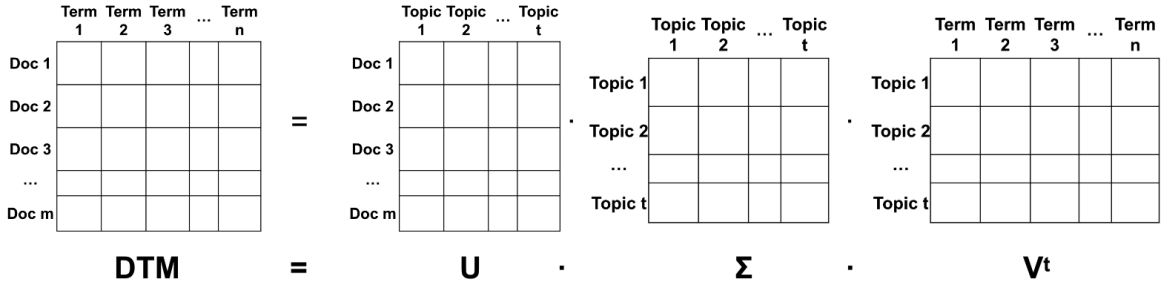
In the process of LSA, SVD decomposes the DTM into the product of three different matrices, shown in the following equation:

$$DTM = U\Sigma V^{\top}$$

In the above equation, the **U** matrix is known as the **Document-topic matrix**, which has the size of m*t. In this part, m means the number of documents, and t means the number of topics.
And the **V** matrix is known as the **Topic-term matrix**, which has the size of t*n. In this part, n means the number of terms, and t means the num of topics[13].

And the **Σ** matrix will be a diagonal matrix that contains singular values from the DTM, and LSA will consider each singular value as a potential topic found in the documents. The size of sigma matrix is t*t. In this part, t means the num of topics.

The computational process is first, LSA selects the first largest singular values ($t \leq \min(m, n)$) of the DTM, and thus discards the last 2 columns (($m - t$) and ($n - t$)) of **U** and **V**, respectively. The rank **t** approximation of the DTM is optimal because it is the closest rank **t** matrix to DTM in terms of **L_2 norm**[16]. By repeating this process, the dimension of the matrix can be reduced. This procedure is known as **truncated SVD,** as sketched in the image below. The resulting approximation of the DTM has rank **t.**



3. State-of-the-art

3.1 SOTA Approach description:

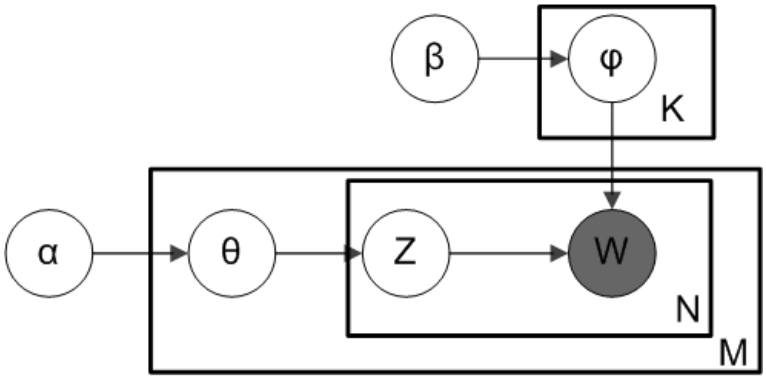
In this section, our team will present two classic State-of-the arts known as: **Latent Dirichlet Allocation** and **Non-Negative Matrix Factorization**.

3.1.1 Latent Dirichlet Allocation

3.1.1.1 Introduction of LDA

Latent Dirichlet Allocation(LDA) is an state-of-the-art appoarch for solving topic modeling problems upon improving **LSA** model. **LDA** assumes input corpus/documents, which are collections of words, are Dirichlet distributed and each of the words will contribute towards the document's topic. **LDA** will search for the best possible matches of the words and topics and output those matches as the topics for the given documents/corpus. [20]

We will introduce the schematics and notations of **LDA**:



M is the total number of corpus

N is the number of words in a given corpus

K is the number of latent topics

α is the parameter of the document-topics Dirichlet distributions

β is the parameter of topic-word Dirichlet distribution

θ_i is the topic multinomial distribution for document i

ϕ_k is the word multinomial distribution for topic k

z_{ij} is the topic sampling from the given topic multinomial distribution θ_i

ω_{ij} is the specific word sampling from the word multinomial distribution ϕ_k given the correspoing topic z_j.

3.1.1.2 Mathematics of LDA

LDA takes a different approach than **LSA**, in which it is a probability-based approach rather than pure linear algebra decompositions. **LDA** assumes that the documents/corpus follow the Dirichlet distribution, thus given the documents and number of topics we want to find, we could cluster each document to the corresponding topic with a certain probabilistic confidence measurements (i.e. **Dirichlet distribution of documents over topics[19]).** Similarly, with topics following the Dirichlet distribution, **LDA** can cluster each topic to the corresponding terms/words with certain pobablistic confidence measurements (i.e. **Dirichlet distribution of topics over terms[19]**). We can view these probabilities as weights for assigning each word to topics and each topic to documents. Given these weights, we then can classify what terms/words belong to a topic, and what topics belong to a single document. The resulting **LDA** probablistic formula is written as below:

$$P(W, Z, \theta, \phi; \alpha, \beta) = \prod_{j=1}^{K} P(\phi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\phi z_{j,t})$$

$P(W, Z, \theta, \phi; \alpha, \beta)$ denotes to 'Total probability of the LDA model'

$P(\phi_i; \beta)$ denotes to 'Dirichlet distribution of topics over terms'

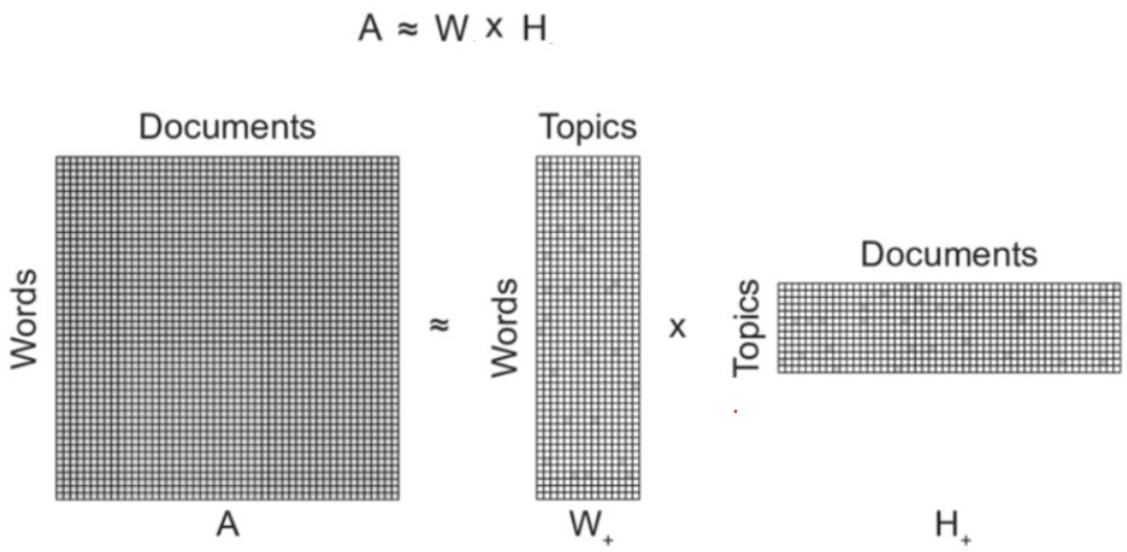
$P(\theta_j; \alpha)$ denotes to 'Dirichlet distribution of documents over topics'

$P(Z_{j,t}|\theta_j)$ denotes to 'prob of a topic appearing given document'

$P(W_{j,t}|\phi z_{j,t})$ denotes to 'prob of word appearing given a topic'

3.1.2 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (**NMF**) is one of the unsupervised clustering approaches to solve the topic modeling problem. The underlying general mathematics is described as below:



Given a term-document matrix $\mathbf{A}_{n\times m}$, where n represents the number of terms/words, and m represents the number of documents, and inside the matrix $\mathbf{A}_{n\times m}$, each value a_{ij} is the TF-IDF value for each term/word in the corpus. The NMF algorithm takes the input $\mathbf{A}_{n\times m}$ and decomposes the matrix into two sub-matrices $\mathbf{W}_{n\times k}$ and $\mathbf{H}_{k\times m}$ where k represents the number of topics and the value of $k \in min(n, m)$. In other words, $\mathbf{W}_{n\times k}$ can be interpreted as given each topic, terms/words that are in the topic and $\mathbf{H}_{k\times m}$ can be interpreted as given each document, topics that are in the document. Therefore, when $\mathbf{W}_{n\times k} \cdot \mathbf{H}_{k\times m}$ each word in $\mathbf{W}_{n\times k}$ will be assigned to a specific weight from $\mathbf{H}_{k\times m}$, indicating the importance of that specific word in a given document, which matches the meaning of each value a_{ij} in matrix[21][22].

NMF procedure is similar to the LSA approach, but with a further improvement on the LSA model. The main difference between LSA and NMF is that the sub-matrices $\mathbf{W}_{n\times k}$ and $\mathbf{H}_{k\times m}$ are always positives (this is not the case with LSA), this implies that the term-document matrix ($\mathbf{A}_{n\times m}$) is also positive for each value. The reason for

such a positive constraint is because negative values in topic modeling are harder to interpret the meaning (i.e. if a word in a topic is assigned to a negative value, it is really hard to tell implication of the word).[21][22]

In order to find $\mathbf{W}_{n\times k}$ and $\mathbf{H}_{k\times m}$, we will need to define an objective function written as below:

$$\frac{1}{2}\|\mathbf{A}-\mathbf{WH}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}(\mathbf{A}_{ij}-(\mathbf{WH})_{ij})^2$$

The objective function is a Frobenius Norm Square function (also known as Euclidean distance), the main goal for this NMF topic modeling task is to minimize the objective function. Then by using the objective rule, the update rule for $\mathbf{W}_{n\times k}$ and $\mathbf{H}_{k\times m}$ can be derived (the derivation can be found here) and the resulting formulas are:

$$\mathbf{W}^{k+1} \leftarrow \mathbf{W}^{k}\frac{(\mathbf{AH}^{\mathrm{T}})}{(\mathbf{WHH}^{\mathrm{T}})}$$

$$\mathbf{H}^{k+1} \leftarrow \mathbf{H}^{k}\frac{({\mathbf{W}^{k}}^{\mathrm{T}}\mathbf{A})}{({\mathbf{W}^{k}}^{\mathrm{T}}\mathbf{W}^{k}\mathbf{H}^{k})}$$

By applying the objective function[22] and the update formulas[22], $\mathbf{W}_{n\times k}$ and $\mathbf{H}_{k\times m}$ can be obtained.

3.2 Why SOTA?

	Pros	Cons
LSA	1. The LSA approach is intuitive[4] 2. Due to the low dimensional representation of the documents. LSA is able to catch the synonyms to some extent.[23]	1. Hard to interpret, since the decomposition contains negative values.[23][24] 2. High computational complexity[23] 3. LSA cannot handle polysemy.[23] 4. The selection of the number of topics is hard for LSA
LDA	1. LDA generally performs better than LSA. 2. Works well with long texts 3. Language agnostic	1. It is a non-deterministic approach, thus this may yield inconsistent results. 2. LDA may result in irrelevant topics. 3. Topics generated are less interpretable than NMF[26][27]
NMF	1. Since it only contains positive values, it is easy to interpret than LSA. 2. It is a deterministic approach, the result is more consistent than LDA 3. NMF performs faster than LDA 4. Works well with short texts 5. Language agnostic 6. NMF can produce a more coherent topic.[24]	1. Initialization is one of the problems for NMF, it is better not to use random initialization for both of the sub-matrices. 2. The objective function might converge slowly or result in a local minimum.

As the above table comparison shows that even though LSA is intuitive and easy to implement, its performance is generally worse than LDA and NMF. Moreover, LSA generates less interpretable results, since it contains mixed

sign values. Whereas, NMF constrains the values to be positives for $\mathbf{W}_{n\times k}$ and $\mathbf{H}_{k\times m}$ which is easier to interpret the results. Besides the interpretations, the computational complexity is also an advantage of both LDA and NMF over the LSA, which is crucial when the input data gets large. Therefore, by justifying the PROS and CONS of the LSA, LDA, and NMF, our team thinks that the state of the arts, LDA and NMF, will be a more preferable choice to LSA for topic modeling problems.

4. Experimental setup:

4.1 Toolkits

Spacy[29]: A well-designed toolkit for Natural Language Processing, it is designed to tag each word with different entities: Nouns, Verbs, Adjectives, etc. Users can actually use it to train their own entities for practical usage. In this project, our team will mainly use its lemmatization tool to reduce various words, which have the same roots but in different forms (e.g. run, runs, running). This will decrease the noise of our dataset when we actually vectorize it. Another usage for Spacy will be tokenization, which will reduce the input text string into arrays of individual words before passing to the vectorization.

Gensim[30]: A toolkit that is specifically designed for topic modeling problems. In this toolkit, there are different models and metrics that we can use for training and evaluation. For instance, LDA and NMF models are already in the toolkit that we can use for implementations. Gensim also provides a Coherence metric that our team can use it for evaluation.

Sklearn[31]: A useful inbuilt library with Python, which contains many statistical and machine learning models. The purpose of using this as our toolkit is because it contains useful functions that we will need for implementation. For instance, vectorize the word using TF-IDF vectorization.

Optimizing and Comparing Topic Model is Simple (OCTIS)[32]: It is an open-source platform, which is built up on **Spacy, Gensim, and Sklearn**. The toolkit is designed to compare different topic models' performance. In the toolkit, it contains an evaluation metric that this tool provides (Topic Diversity) to compare the NMF and LDA with LSA models.

Matlibplot: A well-known plotting tool in Python that we will be used to visualize some of our topic results and exploration of data.

4.2 Logistics

Google Colab: This will be used to implement the code.

GitHub: A platform to store and upload the codes.

Notion: This will be used to write reports.

Google Slides: This will be used for presentation.

4.3 Datasets

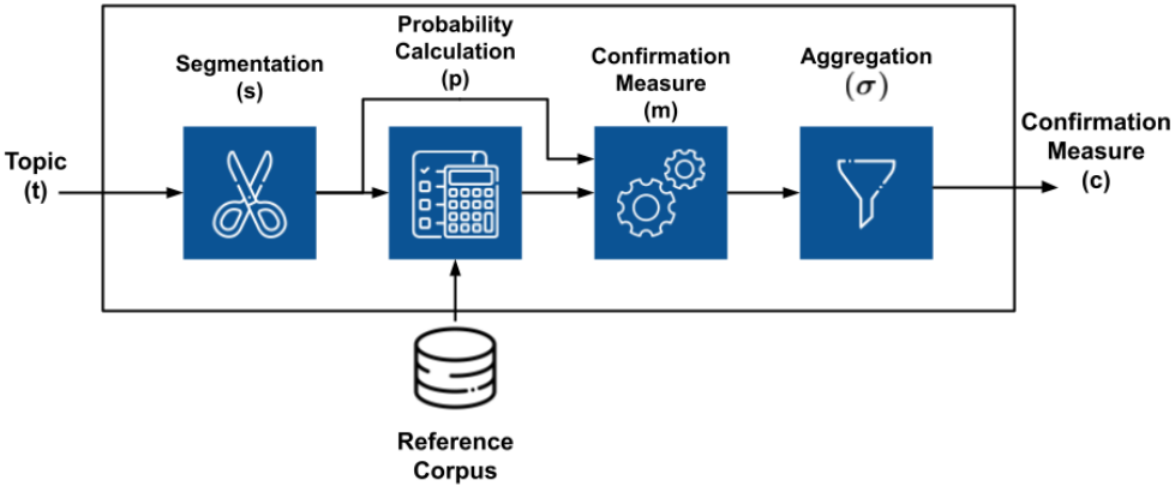
Our group will be using news header datasets: **ABC News Headlines**

ABC News Headlines: The News Headlines has around 1 million news headlines, and the goal of this project is to take these news headlines and find the corresponding topics that will cluster these headlines together by using LDA and NMF models.

4.4 Evaluation Metrics

Coherence: This metric measures topic coherence for the topic modeling. In specific it measures whether the topic texts support their topic [33].

Measuring Topic Coherence consists of four parts indicated below:



The output of the topic models will be segmented into different sets, and calculate the probability of the topic word occurance (in the set) by applying a reference corpus (in our case, we will use default corpus in **Gensim** as our reference corpus). Then the confimation meaure (m) will take different set of topic words and the probability measured in the previous step as inputs; it will output the similarities/relations between each set of the topic words. In the end, it will aggregate these confirmation meaurements and calculate the average of them.

There are different types of Topic Coherence measurements:

Name	C_V	C_{UMass}	C_{NPMI}
Segmentation	S^{one}_{set}	S^{one}_{pre}	S^{one}_{one}
P. Calculation	$P_{sw(110)}$	P_{bd}	$P_{sw(10)}$
C. Measure	$\tilde{m}_{\cos(nlr)}$	m_{lc}	m_{nlr}
Aggregation	σ_a	σ_a	σ_a

Note that each type of the measurement is following the same procedure but with different methods to do segmentation, probability calculation and confirmation measurement. In our experiment we will use C_{UMass} as our topic coherence measurement method. The UMass coherence measurement follows the equation:

$$C_{UMass}(w_i, w_j) = \log \frac{P(w_j, w_j) + 1}{P(w_i)}$$

where $P(w_j, w_j)$ is the word occurrence for both w_i and w_j in the reference corpus. $P(w_i)$ is the word occurance for just w_i in the reference corpus. Thus, the higher the value of the coherence score meaning that the better the texts support the topic [34].

Diversity: This metric indicates the predicted top-k topics' diversities. To measure the diversity, the **OCTIS** toolkit will take the predicted topics and measure the distinction of each predicted topic. The fewer repetitive topic words occur, the more diverse the predictions are. In general, we want our topic models to cover the total amount information available in the training dataset.

---

5. Conclusion

5.1 Next Step

Our next steps:

1. Explore data on **ABC News Headlines**
2. Preprocess the data to obtain the document-term matrix $\mathbf{A}_{n \times m}$
3. Apply **LSA**, **LDA** and **NMF** models on the preprocessed data
4. Obtain topics for each method
5. Compare the topics/models using evaluation metrics

5.2 Expected outcomes

There are two expected outcomes:

1. We would expect **LDA** and **NMF** to perform better than **LSA** in topic modeling task
2. Since the dataset is short texts, we will expect that **NMF** generates more accurate and meaningful topics than **LDA**

---

6. References

[1]"Introduction to Topic Modeling,"*MonkeyLearn Blog*, Sep. 26, 2019. https://monkeylearn.com/blog/introduction-to-topic-modeling/

[2]J. Xu, "Topic Modeling with LSA, PSLA, LDA & lda2Vec,"*Medium*, Dec. 20, 2018. https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

[3]H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, and W. Buntine, "Topic Modelling Meets Deep Neural Networks: A Survey." arXiv, Feb. 28, 2021. Accessed: Oct. 28, 2022. [Online]. Available: http://arxiv.org/abs/2103.00498

[4]S. Bauer, A. Noulas, D. Ó. Séaghdha, S. Clark and C. Mascolo, "Talking Places: Modelling and Analysing Linguistic Content in Foursquare," 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, 2012, pp. 348-357, doi: 10.1109/SocialCom-PASSAT.2012.107.

[5]B. Chen, L. Zhu, D. Kifer, and D. Lee, “What Is an Opinion About? Exploring Political Standpoints Using Opinion Scoring Model,” *AAAI*, vol. 24, no. 1, pp. 1007–1012, Jul. 2010, doi: 10.1609/aaai.v24i1.7717.

[6]Lu, H.-M. and C.-H. Lee, The Topic-Over-Time Mixed Membership Model (TOT-MMM): A Twitter Hashtag Recommendation Model that Accommodates for Temporal Clustering Effects. IEEE Intelligent Systems, 2015(1): p. 1-1.

[7]Xiao, C., et al. Adverse Drug Reaction Prediction with Symbolic Latent Dirichlet Allocation. in AAAI. 2017.

[8]McInerney, J. and D.M. Blei. Discovering newsworthy tweets with a geographical topic model. in NewsKDD: Data Science for News Publishing workshop Workshop in conjunction with KDD2014 the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2014.

[9]Gethers, M. and D. Poshyvanyk. Using relational topic models to capture coupling among classes in object-oriented software systems. in Software Maintenance (ICSM), 2010 IEEE International Conference on. 2010. IEEE.

[10]Linstead, E., et al. Mining concepts from code with probabilistic topic models. in Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. 2007. ACM.

[11]Rao, Y., Contextual sentiment topic model for adaptive social emotion classification. IEEE Intelligent Systems, 2016. 31(1): p. 41-47.

[12]Chen, S.-H., et al. Latent dirichlet allocation based blog analysis for criminal intention detection system. in Security Technology (ICCST), 2015 International Carnahan Conference on. 2015. IEEE.

[13] Dian I. Martin, Michael W. Berry. Mathematical Foundations Behind Latent Semantic Analysis (2013). In T. K. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (pp. 35–55). Lawrence Erlbaum Associates Publishers.

[14] Hamdaoui Y, TF(Term Frequency)-IDF(Inverse Document Frequency) from scratch in python (2019), Towards Data Science

[15] loana, Latent Semantic Analysis: intuition, math, implementation (2020), Towards Data Science

[16] Nicolo Cosimo Albanese, Topic Modeling with LSA, pLSA, LDA, NMF, BERTopic, Top2Vec: a Comparison (2022), Towards Data Science

[17] MonkeyLearn.org. 2022. Topic Modeling: An Introduction. [online] Available at: <https://monkeylearn.com/blog/introduction-to-topic-modeling/>

[18] En.wikipedia.org. 2022. *Latent semantic analysis - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Latent_semantic_analysis>.

[19]Ioana, “Latent Dirichlet Allocation: Intuition, math, implementation and visualisation,”*Medium*, Sep. 26, 2020. https://towardsdatascience.com/latent-dirichlet-allocation-intuition-math-implementation-and-visualisation-63ccb616e094 (accessed Nov. 10, 2022).

[20]R. Kulshrestha, “Latent Dirichlet Allocation(LDA),”*Medium*, Jul. 03, 2020. https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2

[21]C. Goyal, “Topic modelling using NMF: Guide to master NLP (part 14),” *Analytics Vidhya*, 26-Jun-2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf/. [Accessed: 07-Nov-2022].

[22]“Non-Negative Matrix Factorization | Topic Modelling using NMF,” *www.youtube.com*. https://www.youtube.com/watch?v=F0nQHfhMDTU (accessed Nov. 08, 2022).

[23]L. Deng and Phits, “PROBABILISTIC LATENT SEMANTIC ANALYSIS Revised from slides of Shuguang Wang CS3750 Outline • Review of previous notes • PCA/SVD • HITS • Latent Semantic Analysis • Probabilistic Latent Semantic Analysis • Applications Limitations of Probabilistic Latent Semantic Analysis CS3750.” Accessed: Nov. 10, 2022. [Online]. Available: https://people.cs.pitt.edu/~milos/courses/cs3750-Fall2014/lectures/class11.pdf

[24]N. C. Albanese, “Topic Modeling with LSA, pLSA, LDA, NMF, BERTopic, Top2Vec: a Comparison,” *Medium* , Sep. 22, 2022. https://towardsdatascience.com/topic-modeling-with-lsa-plsa-lda-nmf-bertopic-top2vec-a-comparison-5e6ce4b1e4a5 (accessed Nov. 08, 2022).

[25]C. Mehdi, “LDA, NMF, Top2Vec, and BERTopic. How do they work?,”

Medium, Dec. 05, 2021. https://medium.com/@mehdi_chebbah/lda-nmf-top2vec-and-bertopic-how-do-they-work-d71b6365a3d7

[26]R. Chawla, “Topic Modeling with LDA and NMF on the ABC News Headlines dataset,”

Medium, Jul. 30, 2017. https://medium.com/ml2vec/topic-modeling-is-an-unsupervised-learning-approach-to-clustering-documents-to-discover-topics-fdfbf30e27df

[27]“Course:CPSC522/A Comparison of LDA and NMF for Topic Modeling on Literary Themes - UBC Wiki,”*wiki.ubc.ca*

https://wiki.ubc.ca/Course:CPSC522/A_Comparison_of_LDA_and_NMF_for_Topic_Modeling_on_Literary_Themes (accessed Nov. 10, 2022).

[28] not used yet need review:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9120935/#:~:text=BERTopic%2C similar to Top2Vec%2C differs,output the most important topics.

[29]“spaCy · Industrial-strength Natural Language Processing in Python,” *spaCy*, 2015. https://spacy.io/

[30]“Gensim: topic modelling for humans,”*radimrehurek.com*. https://radimrehurek.com/gensim/index.html

[31]scikit-learn, “scikit-learn: machine learning in Python,”*Scikit-learn.org*, 2019. https://scikit-learn.org/stable/

[32]“OCTIS : Optimizing and Comparing Topic Models is Simple!,”*GitHub*, Nov. 07, 2022. https://github.com/MIND-Lab/OCTIS#implement-your-own-model

[33]J. Pedro, “Understanding Topic Coherence Measures,” *Medium*, Jan. 10, 2022. https://towardsdatascience.com/understanding-topic-coherence-measures-4aa41339634c

[34]E. Zvornicanin, “When Coherence Score is Good or Bad in Topic Modeling? | Baeldung on Computer Science,”*www.baeldung.com*, Dec. 07, 2021. https://www.baeldung.com/cs/topic-modeling-coherence-score