Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2025, 2025:11

https://doi.org/10.28919/cmbn/8932

ISSN: 2052-2541

TOPIC MODELING FOR USER FEEDBACK DATASET

SINTA SEPTI PANGASTUTI[1,*], ENENG NUNUZ ROHMATULLAYALY[2], NUROH NAJMI[3]

[1]Department of Statistics, Padjadjaran University, Bandung 45363, Indonesia

[2]Department of Biology, Padjadjaran University, Bandung 45363, Indonesia

[3]Department of Oral Biology, Padjadjaran University, Bandung 45363, Indonesia

Abstract: In the era of big data, user feedback from mobile applications provides valuable insights for improving performance and user experience. However, extracting meaningful topics from large textual datasets remains a challenge. This study employs the Top2Vec model, a modern topic modeling technique, to analyze a dataset containing 15,000 user feedback entries from 15 different mobile applications across various categories. Unlike traditional methods like Latent Dirichlet Allocation (LDA), Top2Vec integrates word embeddings and clustering algorithms to identify topics based on semantic relationships. The research involves text preprocessing, embedding generation using Doc2Vec, and applying the Top2Vec algorithm to extract relevant topics. Results indicate that Top2Vec automatically determines topic numbers, offering richer and more interpretable topics compared to LDA and Embedded Topic Model (ETM). Evaluation metrics such as Coherence Score and Topic Diversity demonstrate that Top2Vec performs well, capturing significant patterns and addressing user concerns, including app glitches, performance issues, and user experience. This article highlights the effectiveness of Top2Vec in analyzing user feedback, making it a promising tool for understanding user needs and improving application development.

Keywords: topic modeling; Top2Vec; user experience; user feedback.

2020 AMS Subject Classification: 62H30.

[*]Corresponding author

E-mail address: sinta.septi@unpad.ac.id

Received September 30, 2024

1. INTRODUCTION

In today's information era, text data has become one of the richest sources of data, whether in the form of news articles, product reviews, social media posts, or research documents. It provides valuable insights that can support strategic decisions in various fields. However, due to the volume and complexity of text data, it is often difficult to manually identify relevant topics or patterns. Therefore, automated approaches to analyzing and understanding text data are becoming increasingly important. Topic modelling is a technique used in Natural Language Processing (NLP) to extract hidden topics from a collection of documents. It is an unsupervised learning technique that aims to discover hidden structures in the form of topics in documents. In unsupervised learning there are three types of clustering: hard clustering, hierarchical clustering, and soft/fuzzy clustering. Topic modeling is included in soft/fuzzy clustering, where each object can have more than one cluster at a given level. The concept of topic modeling according to [1] consists of entities namely "words", "documents" and "corpus". A "word" is considered as a basic unit of discrete data in a document and is defined as an element of vocabulary which is indexed for each unique word in the document. A "document" is an arrangement of N words, and a corpus is a collection of M documents and corpora is the plural of corpus. Meanwhile, "topic" is a distribution of several fixed vocabularies. Topic modeling represents each document as a complex combination of multiple topics.

Conventional approaches such as Latent Dirichlet Allocation (LDA) [2] have been the primary methods for this analysis. It is a probabilistic generative model, that used a mixture of topics and each topic as a mixture of words. In addition, LDA uses bag-of-words (BOW) representations of documents, which ignore word semantics. Another model that combines the power of LDA with word embeddings is Embedded Topic Model (ETM) [3]. This approach overcomes the limitations of the BOW representation of documents and enables a richer semantic representation of words in a document. As a modern alternative, Top2Vec [4] has emerged as a topic modeling technique that combines a representation based on word embeddings with efficient clustering algorithms. Unlike traditional methods, Top2Vec does not rely solely on word frequency but also leverages the semantic meaning contained in the text. By using embedding models such as Universal Sentence Encoder (USE) or Doc2Vec [5], Top2Vec maps documents and words into a vector space where documents with similar contexts are naturally grouped. This approach has been shown to produce more informative and interpretable topics.

Based on [6] that comparing LDA, BERTopic and Top2vec, the results show that it produces 10, 6, and 10 topics related to travel, respectively. However, LDA has 4 topics that were not related to travel. This research proves that Top2vec succeeds in providing more topics. The next study used the LDA and Top2vec models to search for cystic fibrosis (CF) on Reddit. This study uses coherence scores to search and rank the topics most frequently found on the keyword cystic fibrosis (CF). The LDA model produces 9 highest topics and Top2vec produces 68 higher topics, where Top2vec gets a broader topic. This might be due to the absence of stop words in the Top2vec model [7].

Top2vec model proposed by [4] has advantages in its use, besides it automatically detects topics in the text, it no longer needs to use stop words, lemmatization and stemming, which simplifies and expands the search for topics. Another advantage of the Top2vec model is that it can generate topic, document and word vectors embedded in each other such that the distance between them represents semantic similarity. Topic vectors allow calculating topic size based on the closest topic vector of each document vector. In addition, topic reduction can be performed on topic vectors to group similar topics hierarchically and reduce the number of topics discovered. On the other hand, using Top2vec can detect topic vectors in semantic embeddings that represent the main topic. In this study, we analyzed user feedback dataset using Top2vec model to find relevant topic and examine the hidden topic extraction, in order to better understand user’s needs for improvement.

2. MATERIALS AND METHODS

A. Dataset

This study utilizes secondary dataset, comprises user feedback gathered from 15 diverse mobile applications spanning various categories [8]. The data collected included social media, gaming, media streaming, video editing, and payment applications. For each application gathered 1,000 user reviews, resulting in a comprehensive dataset of 15,000 data points from 2018 until 2023. Besides the review content, it includes additional columns such as the score or rating, and app_name. In table 1, five examples from dataset shown below.

Table 1. Five Examples User Feedback Dataset

Review_id	Content	Score	App_name
1_1	Ever since the update, there's a weird glitch where you have to turn on subtitles for every single video….	3	TikTok
2_1	I'm not sure what's been happening. Seems update after update just causes one bug on top of another….	2	Instagram
3_295	I'm sorry for giving only 1 star but I have tried and tried and tried since Christmas to change my profile picture and no freaking matter how I go about it,…	1	Facebook
4_85	Was very simple and easy to use, however with the new update, the split screen is absolutely horrible…	1	WhatsApp
5_167	Best messaging app you can ever imagine….	5	Telegram

B. Text Preprocessing

NLP models work by finding relationships between language constituents, such as letters, words, and sentences in text data. Therefore, before building the model, text data requires preprocessing to improve model performance or change words and characters into a form that the model can understand. Text cleaning process involves removal of unicode characters, lowercasing the text, removing URLs, eliminating numbers and special characters, and removing extra whitespace, therefore only letters remain in the text. After the initial text cleaning process, tokenization is required which allows sentences to be broken down into smaller words and characters or into tokens [9].

The data conversion phase is the phase of adapting the data format used with the aim of simplifying the process or the next phase of text mining. Documents collected during data collection can be obtained in different forms, so in this phase all data is converted into a format that complies with the established text mining data format standards. This research uses files with the xlsx extension, which means that for the data that will be processed later, the Excel format will be used. For the data processing phase, data in the form of strings is used because the Top2vec model can only read data in the form of strings [4]. The data section used for topic modeling analysis uses the “Content” column.

Word embedding is the process of converting text into a vector or array consisting of a collection of numbers [10]. Most machine learning algorithms and deep learning architectures require word embedding because they cannot perform analysis processes on input data in the form of strings or text. Therefore, word embedding is required to convert text as input into numbers. For example, when a machine learning model receives text as input, machine learning cannot accept the raw text directly. Machine learning reads the text by creating a word dictionary that contains all the words present in the dataset. Then, every time a word sequence is received, the string is converted into an integer by assigning a number to it. This numbering can be determined by the order in the word dictionary we have.

C. Top2Vec Model

Top2vec is a topic modeling and semantic search model. This model automatically detects the topics present in the text and generates topics, documents and word vectors embedded along with the documents. It supports the use of embedding models and can generate topic, document and word vectors that are merged so that the distance between them represents semantic similarity [4]. The output generated using Top2vec is as follows:

a. Gets the number of topics detected.

b. Gets the topic and size of the topic.

c. Search topics using keywords.

d. Search documents by topic.

e. Search documents by keyword.

f. Find similar words.

g. Similar documents found

1. Top2Vec Algorithms

   The algorithm assumes that many semantically similar documents point to an underlying topic. The first step is to create a joint embedding of document and word vectors. Once documents and words are embedded in a vector space, the goal of the algorithm is to find dense clusters of documents and then identify which words these documents collectively attracted. Each dense region is a topic and the words that attracted the documents to the dense region are the topic words. Here is the step by step from Top2vec algorithm:

   a. Create jointly embedded document and word vectors using Doc2Vec or Universal Sentence Encoder or BERT Sentence Transformer.

   b. Create an embedding of document vectors in lower dimensions using UMAP.

   c. Find dense regions of documents using HDBSCAN.

   d. For each dense region, compute the centroid of the document vectors in the original dimension, this is the topic vector.

e. Find n-closest word vectors to the resulting topic vector.

2. Parameter Model

According to [4], the Top2vec model has three parameters which are: Fast_Learn, Learn and Deep_Learn. This parameter determines how fast the model performs training. The fast_learn option is the fastest parameter and produces the lowest quality vector. The learn option performs training and produces better quality vectors but takes longer time. The deep_learn option performs training and produces the best quality vectors but requires a significant and longer time compared to other parameters. However, there is an iteration in the Top2vec parameter called Workers parameter.

3. Processing data

To obtain quality top2vec modeling, several methods are required: pre-trained embedding models, training models and determining the number of iterations using coherence to build the best model to use, as well as calculating the value of each word found using topic gain information.

a. Pre-trained embedding model

The Top2vec model is equipped with Doc2Vec, which is used by default to generate a combination of words and document embeddings. Since the data used is 15,000 data, the Do2vec embedding model is used to obtain a larger topic.

b. Training model

The first step is to convert the data into lists and strings so that it can be read by Top2vec and then proceed with document and word vector embedding. After obtaining the documents and words, the research involved taking the headline data from each news story to reduce the data size and find a group of documents with a large amount of data. Then, an embedding vector, namely Doc2Vec is used while training the Top2vec model to see how many topics are generated. This test uses 3 parameters of Top2vec: Fast-Learn, Learn and Deep-Learn to get topics. Then, dimension reduction is done using Uniform Manifold Approximation & Projection (UMAP) [11] and HDBSCAN [12]. After calculating by the model and giving the number of topic results obtained from the document. The results obtained from the Top2vec model are semantically similar documents that give a hint about the underlying topic.

c. Coherence score

Coherence is an evaluation metric that can be used to assess the performance of the topic model [13]. Coherence is typically used to analyze the relationship between two sets of data or the similarity between data sets. In topic modeling, topic coherence measures the quality of the data by comparing the semantic similarity between highly repetitive words in a topic [14]. Determining the number of iterations using a multi-topic experiment is the step that generates a model with a coherence value. The higher the coherence value, the better the model accuracy.

D. Metrics Evaluation

To evaluate the quality of topic models based on topic coherence (topic keywords must shame some level of semantic relatedness) and topic segregation, which measures the lexical and semantic overlap between topics. Note that a higher value indicates a better performance in all the metrics mentioned below.

a. Coherence UCI developed by [15], it quantifies the coherence of a topic by measuring the pointwise mutual information (PMI) between the top words in the topic.

$$UCI = \frac{2}{N\cdot(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathrm{PMI}\left(w_i, w_j\right) \quad (1)$$

N represents the number of top words in a topic. The PMI between two words is defined below:

$$\mathrm{PMI}\left(w_i, w_j\right) = \log \frac{P\left(w_i, w_j\right) + \varepsilon}{P(w_i).P\left(w_j\right)} \quad (2)$$

$P(w_i)$ and $P\left(w_j\right)$ represent the probabilities of observing words w_i and w_j, respectively. $P\left(w_i, w_j\right)$ calculates the probability of co-occurrence of two words, and the terms ε is a small constant added to the probabilities to avoid issues with zero probabilities.

b. Normalized Pointwise Mutual Information (NPMI) [16] is one of the most well-known automatic coherence metrics. It measures how much more likely the most representative terms of a topic co-occur than if they were independent. The range of NPMI is between [-1,1], offering a more comprehensive evaluation of topic coherence.

$$\mathrm{NPMI}(\mathrm{w_i}, \mathrm{w_j}) = \frac{\log \frac{P(w_i,w_j)+\varepsilon}{P(w_i).P(w_j)}}{-\log\left(P(w_i,w_j)\right)+\varepsilon} \tag{3}$$

c. Coherence CV [17] uses a variation of NPMI to calculate the coherence over sliding window with size 110. Unlike the previous coherence metrics, CV does not directly use co-occurrence frequency but rather focuses on the co-occurrence of top words with other top words, allowing for greater sensitivity to the semantics of the topic. It calculates the co-occurrence of a word of a given topic against all words of the same topic. The CV score is based on the NPMI score, an advanced way to calculate the probability of two words co-occurring in a corpus.

$$\cos(\vec{u}, \vec{w}) = \frac{\sum_{i=1}^{|W|} u_i \cdot w_i}{\|\vec{u}\|_2 \cdot \|\vec{w}\|_2} \tag{4}$$

d. Topic Diversity [18] asses the diversity of topics based on the unique words present in the topic-word distributions. It is measured how distinct each topic is from others, instead of focusing on the coherence of words within topics. The calculation for topic diversity (*td*) as follows:

$$td = \frac{|unique\_words|}{w \times |topics|} \tag{5}$$

3. Main Results

A. Model Comparison

Top2vec model has three parameters used to train the model to get maximum topics. The parameters used in this model are Fast-Learn, Learn and Deep-Learn, which give different topics.

Table 2. Key Feature

Model Embedding	all-MiniLM-L6-v2
Topic Span	C-Top2Vec automatically determines the number of topics and finds topics segments within documents, allowing for a more granular topic discovery

After preprocessing the dataset, we conduct each model with Feedback dataset, as shown ini Table 3.

Table 3. Coherence Score for Topic Modeling

Model	Coherence Score			Topic
	CV	NPMI	UCI	Diversity
Top2Vec	0.40968	-0.18416	-5.8157	0.16419
LDA	0.40608	-0.00526	-0.3075	0.6
ETM	0.42360	-0.00737	-0.2552	0.37142

From Table 3, it is shown that a score comparison for each model from Coherence CV, Coherence NPMI, Coherence UCI and Topic Diversity explain the ability of the model to measures the quality of the data by comparing the semantic similarity between highly repetitive words in a topic. As we know, a higher value indicates a better performance in all the metrics mentioned above.

B. Topic Insights

In the following discussion, we will delve more into the topics generated by the Top2vec model. The resulting six topics are distinguishable and relevant. For the topic keywords of each topic, refer to Table 4.

Table 4. Topic Keywords Generated

Topic	Topic Keywords
1	['app', 'ios', 'youtube', 'facebook', 'android', 'mobile', 'instagram', 'application', 'netflix', 'newsfeed', 'ui', 'usability', 'streaming', 'smartphone', 'homescreen', 'cashapp', 'spotlight', 'functionality', 'activity', 'content', 'roku', 'podcast', 'tablet', 'viewer', 'ipad', 'feed', 'spotify', 'twitter', 'promote', 'iphone', 'suggestion', 'homepage', 'recommendation', 'screen', 'youtuber', 'messenger', 'software', 'curate', 'entertain', 'sharing', 'stream', 'aim', 'entertainment', 'chatbot', 'entertaining', 'internet', 'whatsapp', 'snapchat', 'google', 'alternative']
2	['tiktok', 'android', 'app', 'reinstall', 'ios', 'uninstal', 'uninstall', 'tik', 'crashing', 'screen', 'mobile', 'glitchy', 'tablet', 'ipad', 'reboot', 'reload', 'issue', 'unable', 'restart', 'troubleshoot', 'glitch', 'youtube', 'refresh', 'troubleshooting', 'homescreen', 'crash', 'reset', 'bug', 'smartphone', 'videocall', 'laggy', 'sluggish', 'redownload', 'try', 'lagging', 'fullscreen', 'problem', 'deactivate', 'tok', 'uninstalling', 'disable', 'messenger', 'update', 'application', 'spotify', 'trouble', 'wallpaper', 'samsung', 'fix', 'buggy']

3	['laggy', 'lagging', 'videocall', 'lag', 'video', 'streaming', 'youtube', 'sluggish', 'playback', 'crashing', 'loading', 'delay', 'reload', 'flicker', 'stream', 'youtuber', 'troubleshooting', 'reinstall', 'vid', 'troubleshoot', 'unresponsive', 'glitchy', 'slow', 'issue', 'unable', 'responsive', 'bandwidth', 'refresh', 'blurry', 'redownload', 'fullscreen', 'reboot', 'autoplay', 'load', 'restart', 'android', 'viewer', 'homescreen', 'screen', 'chromecast', 'bug', 'glitch', 'problem', 'camera', 'mobile', 'crash', 'faulty', 'uninstall', 'uninstal', 'virus']
4	['login', 'password', 'reinstall', 'account', 'reset', 'deactivate', 'authentication', 'activate', 'unable', 'app', 'uninstal', 'issue', 'uninstall', 'token', 'gmail', 'log', 'reconnect', 'reactivate', 'unlock', 'logout', 'register', 'reboot', 'problem', 'message', 'hacker', 'error', 'ban', 'verification', 'email', 'android', 'messenger', 'restart', 'mobile', 'delete', 'troubleshoot', 'unsubscribe', 'hack', 'trouble', 'disconnect', 'restore', 'access', 'reload', 'uninstalling', 'application', 'troubleshooting', 'messanger', 'reappear', 'ios', 'subscription', 'successfully']
5	['youtube', 'youtuber', 'video', 'videocall', 'vid', 'homescreen', 'playback', 'reload', 'bug', 'reinstall', 'laggy', 'android', 'autoplay', 'crashing', 'streaming', 'chrome', 'screen', 'uninstal', 'issue', 'uninstall', 'lagging', 'fullscreen', 'mobile', 'refresh', 'browser', 'redownload', 'chromecast', 'pause', 'disable', 'app', 'reboot', 'viewer', 'glitchy', 'restart', 'netflix', 'flash', 'responsive', 'flicker', 'fixing', 'sluggish', 'update', 'glitch', 'watch', 'tv', 'fix', 'stream', 'malfunction', 'unable', 'playlist', 'deactivate']
6	['wifi', 'reconnect', 'reinstall', 'connection', 'connectivity', 'disconnected', 'app', 'reload', 'disconnect', 'connect', 'unable', 'restart', 'reboot', 'loading', 'troubleshooting', 'mobile', 'troubleshoot', 'laggy', 'sluggish', 'refresh', 'uninstal', 'offline', 'uninstall', 'reset', 'issue', 'lagging', 'error', 'problem', 'android', 'unresponsive', 'crashing', 'network', 'ios', 'internet', 'smartphone', 'router', 'unavailable', 'unreliable', 'lag', 'deactivate', 'provider', 'paywall', 'load', 'bandwidth', 'ping', 'retry', 'redownload', 'glitchy', 'connected', 'unstable']

To visualize the distribution of topics generated by the Top2vec model, refer to Figure 1, which presents a bar plot of the topic frequencies. We can observe that the model performs well, as the topics are evenly generated, avoiding any strong imbalance. The most frequent topic is Topic 53, with 0.9277 probability, while the least frequent is Topic 42, with 0.9136 probability.

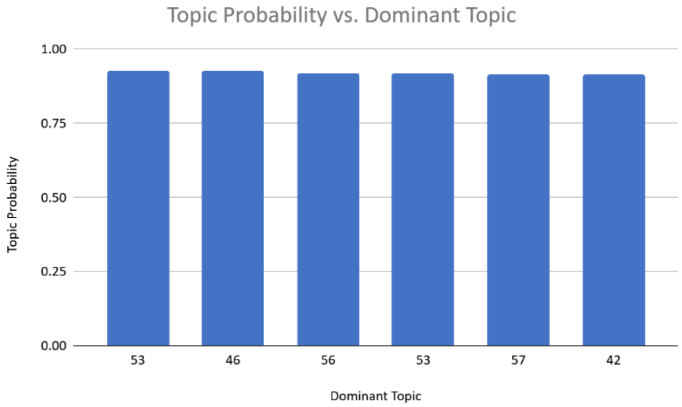

Figure 1. Topic distribution of the topics generated by the Top2vec model

For the topic keywords of each topic, refer to Table 3, we will now provide more detailed insights into each topic:

- Media social and streaming platform: This topic is related to media social to connect with people, also music, audio streaming, playlists, premium features, and podcasts.
- App Troubleshooting on TikTok and Spotify: This topic revolves around user feedback related to app performance issues, including phone restarts, uninstallation, and dealing with crashes.
- App Troubleshooting on YouTube and others video streaming: This topic revolves around user feedback related to app performance issues, including phone restarts, uninstallation, and dealing with crashes.
- User Experience: This topic seems to show user experienced on uninstall, unsubscribe or even successfully dealing with the troubling in application.
- Video streaming platform: This topic mainly describes YouTube, videocall, Netflix.
- App Troubleshooting in general.

C. Conclusion

In conclusion, this study analyzes and extracts meaningful insights from vast amounts of unstructured user feedback data using Top2Vec model. The topics generated by our model provide insights into various aspects of user experiences and opinions. The topics include ‘‘Media social and streaming platform’’, ‘‘App Troubleshooting’’, ‘‘User Experience’’, ‘‘Video streaming platform’’, and ‘‘App Troubleshooting in general’’. Looking ahead, there are several potential future developments that can further enhance the field of user feedback analysis. Firstly, researchers can explore the usage of this dataset or variations of it, using the Contextualized Topic Modeling to delve deeper into specific domains or applications.

ACKNOWLEDGMENT

This work was supported by Padjadjaran University, Library and Online Data Research (RDPD) research grant.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet Allocation, J. Mach. Learn. Res. 3 (2003), 993–1022.

[2] H. Jelodar, Y. Wang, C. Yuan, et al. Latent Dirichlet Allocation (LDA) and Topic Modeling: Models, Applications, A Survey, Multimedia Tools Appl. 78 (2019), 15169–15211. https://doi.org/10.1007/s11042-018-6894-4.

[3] A.B. Dieng, F.J.R. Ruiz, D.M. Blei, Topic Modeling in Embedding Spaces, Trans. Assoc. Comput. Linguist. 8 (2020), 439–453. https://doi.org/10.1162/tacl_a_00325.

[4] D. Angelov, Top2Vec: Distributed Representations of Topics, Preprint, 2020. https://arxiv.org/abs/2008.09470.

[5] J.H. Lau, T. Baldwin, An Empirical Evaluation of Doc2vec with Practical Insights into Document Embedding Generation, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), 2014, pp. 1–8.

[6] R. Egger, J. Yu, A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts, Front. Sociol. 7 (2022), 886498. https://doi.org/10.3389/fsoc.2022.886498.

[7] B. Karas, S. Qu, Y. Xu, Q. Zhu, Experiments with LDA and Top2Vec for Embedded Topic Discovery on Social Media Data—A Case Study of Cystic Fibrosis, Front. Artif. Intell. 5 (2022), 948313. https://doi.org/10.3389/frai.2022.948313.

[8] M.H. Asnawi, A.A. Pravitasari, T. Herawan, T. hendrawati, User Feedback Dataset from the Top 15 Downloaded Mobile Applications, in: The Combination of Contextualized Topic Model and MPNet for User Feedback Topic Modeling (1.0.0, Vol. 11, pp. 130272–130286), Zenodo, (2023). https://doi.org/10.5281/ZENODO.10204232.

[9] G. Grefenstette, Tokenization, in: H. Van Halteren (Ed.), Syntactic Wordclass Tagging, Springer Netherlands, Dordrecht, 1999: pp. 117–133. https://doi.org/10.1007/978-94-015-9273-4_9.

[10] S. Olumide, The Key to LLMs: A Mathematical Understanding of Word Embeddings, KDnuggets, 2024. https://www.kdnuggets.com/the-key-to-llms-a-mathematical-understanding-of-word-embeddings.

[11] L. McInnes, J. Healy, J. Melville, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, Preprint, (2020). https://doi.org/10.48550/arXiv.1802.03426.

[12] R.J.G.B. Campello, D. Moulavi, J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates, in: J. Pei, V.S. Tseng, L. Cao, H. Motoda, G. Xu (Eds.), Advances in Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, 2013: pp. 160–172. https://doi.org/10.1007/978-3-642-37456-2_14.

[13] M. Pickett, Exploring Coherence Metrics for Optimizing Topic Models of Humpback Song, MBARI, 2020. https://www.mbari.org/wp-content/uploads/Pickett.pdf.

[14] S. Kapadia, Evaluate topic models: Latent Dirichlet Allocation (LDA), Towards Data Science, 2019. https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0.

[15] D. Newman, J.H. Lau, K. Grieser, T. Baldwin, Automatic Evaluation of Topic Coherence, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, 2010, pp. 100–108. https://dl.acm.org/doi/10.5555/1857999.1858011.

[16] N. Aletras, M. Stevenson, Evaluating Topic Coherence Using Distributional Semantics, in: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers, Potsdam, Germany, 2013, pp. 13–22.

[17] M. Röder, A. Both, A. Hinneburg, Exploring the Space of Topic Coherence Measures, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM, Shanghai China, 2015: pp. 399–408. https://doi.org/10.1145/2684822.2685324.

[18] A.B. Dieng, F.J.R. Ruiz, D.M. Blei, Topic Modeling in Embedding Spaces, Trans. Assoc. Comput. Linguist. 8 (2020), 439–453. https://doi.org/10.1162/tacl_a_00325.