Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 8, No. 1, February, 2024

Topic Modeling: A Consistent Framework for Comparative Studies

Ana Amaro [1*], Fernando Bacao [1]

[1] *NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Portugal*.

Abstract

In recent years, the field of Topic Modeling (TM) has grown in importance due to the increasing availability of digital text data. TM is an unsupervised learning technique that helps uncover latent semantic structures in large sets of documents, making it a valuable tool for finding relevant patterns. However, evaluating the performance of TM algorithms can be challenging as different metrics and datasets are often used, leading to inconsistent results. In addition, many current surveys of TM algorithms focus on a limited number of models and exclude state-of-the-art approaches. This paper has the objective of addressing these issues by presenting a comprehensive comparative study of five TM algorithms across three different benchmark datasets using five different metrics. We offer an updated survey of the latest TM approaches and evaluation metrics, providing a consistent framework for comparing different algorithms while introducing state-of-the art approaches that have been disregarded in the literature. The experiments, which primarily use Context Vectors (CV) Topic Coherence as an evaluation metric, show that Top2Vec is the best-performing model across all datasets, disrupting the tendency for Latent Dirichlet Allocation to be the best performer.

Keywords:

Natural Language Processing;
Top2Vec;
Topic Coherence; Topic Modeling;
Unsupervised Learning.

Article History:

Received:	14	June	2023
Revised:	27	December	2023
Accepted:	09	January	2024
Published:	01	February	2024

1- Introduction

The volume of digitally available text has grown exponentially, rendering manual analysis of this unstructured data an infeasible task. Consequently, techniques that allow for automatic extraction of information, known as information retrieval, have become increasingly important. One such approach that aids in the analysis of this type of data is Topic Modeling (TM), which involves grouping similar data elements, often referred to as documents. This unsupervised learning technique, whose popularity has been on the rise [1], leverages the latent semantic structure of documents to cluster them according to their content while offering insights into their distinguishing characteristics.

With the number of models proposed in the literature increasing, different evaluation mechanisms at the metric and dataset levels are used for their proposals [2]. This diversity of evaluation approaches significantly hinders the consistency and comparability of the different techniques introduced across models and applications. Although recent research has addressed this issue, the literature mostly focuses on either comparing only two models [3-5] or examining very specific contexts [4, 6, 7]. Among the existing studies, the research done by Lisena et al. [2] and Harrando et al. [8] stands out as the most comprehensive to date. They employ two and three datasets, respectively, to evaluate various algorithms representing distinct approaches to TM. Nevertheless, despite the works being recent, the models employed do not reflect the current state-of-the-art, lacking new approaches that have the potential to improve the obtained results.

In this work, we aim to address the aforementioned limitation by incorporating algorithms that have emerged in the meantime, in addition to the ones that had the best performance in the studies of Lisena et al. [2] and Harrando et al. [8]. The algorithms used will be evaluated on the same datasets using the same evaluation metrics. This approach ensures

[*] **CONTACT**: m20200598@novaims.unl.pt

DOI: http://dx.doi.org/10.28991/ESJ-2024-08-01-09

consistency and comparability across experiments, enabling a fair and meaningful assessment of the algorithms' effectiveness for TM.

The remainder of this work is organized into four sections. The next section comprehends the Theoretical Background, including context on TM, its algorithms, the evaluation metrics utilized, and a review of previous surveys. Subsequently, we present the methodology employed in our work, followed by a section dedicated to the description of the experiments conducted and the respective results. Finally, the last section concerns the conclusions drawn from the experiments and offers suggestions for future work.

2- Related Work

Topic Modeling (TM) falls under the domain of Natural Language Processing (NLP), being an unsupervised learning technique used to find latent semantic structures within a collection of documents. The fundamental idea behind this technique is to explore the concept that each document tends to address at least one specific topic, which can be characterized by a meaningful set of words.

Traditionally, TM methods have adopted a generative probabilistic approach, wherein each document is viewed as a mixture of topics, and each topic is represented by a probability distribution of words. Nevertheless, some recent techniques follow a different approach: while each topic is still represented by a probability distribution over words, each document is considered to belong to a single topic. This shift from traditional approaches reflects an evolution in TM techniques and highlights the diversity of methods employed in contemporary research.

TM allows users to perform tasks at a much faster pace while reducing their complexity, making it highly efficient in grouping documents according to their topics on a larger scale than what a human could achieve manually. Due to its utility, it has seen applications in several fields. In the realm of NLP, TM has been employed for sentiment analysis [9], social media event extractions [10], text generation [11], document summarization [12], word sense disambiguation [13], translation [14], and has even been used for computer vision tasks [15]. Additionally, on a more macro-view, TM has been applied to a range of different areas, such as Politics [16], Healthcare [17], Marketing [18], Sociology [19], Economics [20], Employment [21] or Business [22].

Over the years, different techniques have been proposed in the field of TM. The pioneering approach, originally proposed to identify the most similar documents to a user's query, was Latent Semantic Indexing (LSI) [23]. LSI leveraged Single Value Decomposition to extract latent semantic structures from a collection of documents. Subsequently, several algorithms specifically intended for TM were introduced: Probabilistic Latent Semantic Analysis (pLSA) [24], Non-Negative Matrix Factorization (NMF) [25], and, the most well-known and widely used, Latent Dirichlet Allocation (LDA) [26].

Following these initial approaches, several models have been proposed over the years to cater to specific contexts, address limitations, or incorporate new advances in the Text Mining field, namely new document embedding techniques. In this regard, topic models traditionally employed the Bag of Words (BoW) method [27] to embed documents, which only takes into consideration word frequency, leading to the loss of semantic meaning and disregarding word order.

Nonetheless, advances in Deep Learning have led to the emergence of new text embedding methods that address the BoW's limitations, enabling the consideration of semantics and the preservation of word order. Techniques such as Word2Vec [28], Doc2Vec [29], Glove [30], Fast-text [31], Probabilistic Fast-text [32], Bidirectional Encoder Representations from Transformers (BERT) [33], and Sentence-BERT (S-BERT) [34] are examples of these new embedding techniques. Being able to represent text in a more meaningful way allowed for the exploration of new TM models, which took advantage of these text representations to create algorithms that were able to preserve the text's structure and semantics.

Existing surveys in the literature can be broadly categorized into two main groups. Some surveys exclusively focus on traditional models [4, 5, 6, 35]; others combine both traditional and neural models [2, 3, 7, 8, 36]. Besides this, some studies have a more narrow scope, limiting their analysis to only two models [3-5], while others have chosen to address a specific use case [4, 6, 7].

Regarding the surveys' conclusions, even though neural topic models would be expected to outperform the more traditional algorithms proposed several years before, in studies in which both types are included, the opposite happens: LDA tends to obtain the best results [2, 4, 6, 8, 35].

Concerning the metrics employed, some studies rely on 'traditional' metrics such as Perplexity, Recall, Precision, or F1-Score [3, 35] when there is information about the test set. On the other hand, some authors prefer to employ a combination of the aforementioned metrics with TC and human evaluation [2, 6, 7, 8, 36]. Lastly, the studies presented in Mohammed and Al-Augby [4] and O'Callaghan et al. [5] offer us examples of authors who prefer to use only TC to evaluate an algorithm's solution.

The most comprehensive studies on Topic Modeling are the studies of Lisena et al. (2020) [2], and Harrando et al. (2021) [8], which explore two and three datasets, respectively. In these works, the authors apply the following nine models: LDA, Latent Feature Topic Models (LFTM)[*], Doc2Topic (D2T)[†], Gibbs Sampling for a Dirichlet Mixture Model (GSDMM)[‡], NMF, Hierarchical Dirichlet Processing (HDP) [37], LSI, PVTM, and Context Topic Model (CTM)[§].

2-1- Review of the Models Evaluated

In this work, we conduct a comparative analysis of five different models: PVTM [38], Top2Vec [39], BERTopic [40], and the more traditional LDA [26] and NMF [25]. LDA and NMF are included in this work not only for representing the most conventional take on TM, but also for having been previously identified as being among the top three best-performing models in the studies of Lisena et al. [2] and Harrando et al. [8]. The other algorithm present in this top three is PVTM, which is thus also part of this work. To introduce more state-of-the-art approaches, this work includes Top2Vec and BERTopic, in representation of more recent approaches, with the potential to outperform the remaining three models in analysis. The following table (Table 1) summarizes these five algorithms, and their key properties.

Table 1. Comparison of algorithms' key aspects

Algorithm	Authors	Key Aspects
LDA	[26]	Aims at obtaining a corpus approximation, recurring to topic-word and document-topic distributions. Follows a generative probabilistic approach. Uses BoW as an embedding method and requires the number of topics to be defined a priori.
NMF	[25]	Uses a document-term matrix as the product of word-topic and document-topic matrixes. Requires the number of topics to be defined a priori.
PVTM	[38]	Represents documents and words in a semantic space and clusters the former through a Gaussian Mixture Model. Employs Doc2Vec as an embedding technique and requires the number of topics to be known a priori.
Top2Vec	[39]	Represents documents and words in a sematic space and clusters the former through HDBSCAN. Uses Doc2Vec as an embedding technique and does not require the number of topics to be defined.
BERTopic	[40]	Clusters documents through HDBSCAN, finding topic vectors through class-based TF-IDF. Uses BERT as an embedding technique and does not require the number of topics to be defined.

LDA [26] is the most well-known and used model in the field of TM, a reflection of its simplicity of use and tendency for good results. The algorithm utilizes the Dirichlet distribution to model both the topics' distribution over words and the documents' distribution over topics. This distribution becomes sparser as one of its parameters decreases, which allows to obtain a good representation of documents given these sparse identities (e.g., a document will not contain all words in the corpus). LDA aims at generating a good corpus approximation by building the document-term matrix through the topic-word and document-topic distributions. However, LDA utilizes the most frequent words as representatives, which can become problematic if the corpus is not pre-processed due to the existence of stop-words that can dominate the results. Hence, pre-processing is a must when applying LDA to ensure meaningful topic representations. One limitation of LDA is that it requires the number of topics to be known a priori, which is something rarely known in a real-life scenario. Lastly, LDA is not well-suited for short text [41].

NMF, proposed in Paatero & Tappe [25], consists of decomposing a non-negative matrix in the multiplication of two matrixes, both also non-negative, by performing a low-rank approximation. The document-term matrix (with the shape Number of Words in Corpus x Number of Documents in Corpus) can be decomposed into the product of a word-topic matrix (with the shape Number of Words in Corpus x Number of Topics) and a document-topic matrix (with the shape Number of Topics x Number of Documents in Corpus). By minimizing a cost function whose goal is to make sure that the matrix multiplication is as close to the original document-term matrix as possible, NMF aims at finding a local minimum.

Once more, this algorithm demands the number of topics (e.g., the factorization rank) to be specified beforehand while requiring the text to be pre-processed.

Paragraph Vector Topic Model (PVTM) [38] aims at representing documents and words in a semantic space, recurring to Doc2Vec [29] to perform the words and documents' embedding. With the documents' vector representations found, PVTM takes advantage of the fact that in a semantic space, the vectors that are closer are more semantically similar in order to cluster these through a Gaussian Mixture Model.

[*] https://github.com/datquocnguyen/LFTM

[†] https://github.com/sronnqvist/doc2topic

[‡] https://github.com/rwalk/gsdmm

[§] https://github.com/MilaNLProc/contextualized-topic-models

A downside of this clustering technique is the requirement to define the number of clusters in advance (as with LDA and NMF). Once the documents are semantically grouped, the vectors that represent the clusters' centroids—topic vectors—can be computed, representing the 'average' document of each cluster. Afterwards, having both words and documents in the same semantic space, cosine similarity is used to discover the most similar words to each topic vector, hence finding each cluster's most characteristic words.

Top2Vec [39] also refers to Doc2Vec [29] as an embedding technique to create jointly embedded words and documents, preserving the words' semantics. As with PVTM, the main reasoning behind this approach is to take advantage of the semantic space in which both words and documents are represented. By using this embedding technique, Top2Vec is also able to deviate itself from some disadvantages of the BoW approach, namely the loss of semantics.

Considering documents can be represented as vectors with high dimensionality, in order not to incur the Curse of Dimensionality [42], Uniform Manifold Approximation and Projection (UMAP) [43], a manifold learning technique, is the chosen tool to reduce dimensionality, since the author found it to be better than t-distributed Stochastic Neighbor Embedding (t-SNE) [44] at preserving the global structure and when dealing with large datasets.

In this reduced dimension, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [45] is applied to the transformed documents' vectors to discover dense areas, benefiting from the semantic space to find meaningful semantic clusters. As with PVTM, the reasoning is that a dense area of semantically similar documents will represent documents that concern the same topic, allowing for the centroid (topic vector) of each cluster to be a representative document of its elements. Taking advantage of the continuous representation of topics and the joint words and documents' embedding the words closer to each topic vector are considered the most representative ones.

Top2Vec has the advantage of not requiring any pre-processing, namely stop-word removal, lemmatization, or stemming. In addition, this algorithm can find the optimal number of clusters by itself. Lastly, the author claims to have found topics 'significantly more informative and representative of the corpus trained on than those found by LDA and pLSA' [39].

BERTopic [40] builds on Top2Vec to introduce some changes and a more recent embedding technique. The documents' embedding is done recurringly in BERT [33], in detriment of Doc2Vec. The dimensionality reduction and clustering are performed using the same approaches as with Top2Vec: UMAP and HDBSCAN. However, since the words and documents are not projected in the same semantic space, there is a need for a new way to find representative words for each cluster. For this, an adaptation of the Term Frequency – Inverse Document Frequency (TF-IDF) [46] is applied: Class-based TF-IDF. This variation works in the following way: all documents from the same cluster are combined into just one document, leading to the same number of documents and clusters; from this point, TF-IDF can be used as usual, finding the most relevant words within each cluster to differentiate it from the remaining.

As a final step, BERTopic takes advantage of Maximal Marginal Relevance (MMR) [47] to assess the similarity between a document's last chosen descriptive word and both the document itself and the previous words already selected to ensure that only words that will bring diversity are chosen, avoiding redundancy and words that are not discriminative enough throughout topics.

2-2-Evaluation Metrics

The evaluation of a topic model's output, which typically consists of the 10 words that better describe each topic [48, 49], is not a straightforward process. Being a subjective task, the assessment of how good these words are at describing the topic at hand was traditionally done by humans through manual inspection. In this regard, Chang et al. [50] introduced two metrics for evaluating topics' interpretability: topic intrusion and word intrusion. Topic intrusion consisted of, given a document, presenting the associated topics and an extra one that had not been considered related to the document at hand, with the goal being to assess if a person could distinguish what was the non-applicable topic. On the other hand, word intrusion was assessed by, given a topic, displaying the words that best characterized it alongside extra ones. Once more, the objective was to assess if a person could easily detect which words didn't belong next to the remaining. In both cases, the easier the task was for a human controller, the better the solution's quality was offered by the model. Nonetheless, this approach is faulty, at best, as it requires human intervention, which is not only subjective but also time- and cost-consuming.

Intending to improve this, in Newman et al. [49], the authors proposed Topic Coherence (TC), a metric computed without human intervention, whose goal is to mimic the human perspective when assessing if a topic is coherent or not. The authors found that when computing TC through term co-occurrence, based on Pointwise Mutual Information (PMI) (UCI Coherence), the results obtained were in line with human labeling. Building on this work, similar metrics have since been proposed. For instance, in Mimno et al. [51], the researchers have proposed a TC measure through term co-occurrence, but recurring to log conditional probability, while Lau et al. [52] introduced a TC similar to the one in Newman et al. [49], but utilizing Normalized Pointwise Mutual Information (NPMI Coherence) [53] instead and showed that TC is able to obtain the same results as human experiments. On the other hand, Aletras & Stevenson [48] approached

TC differently, claiming to have surpassed previous methods through 'distributional semantic similarity methods', by representing each topic's words as a vector in a semantic space and creating context vectors for each word based on word co-occurrence. Then, TC was obtained from either the similarity between the words' context vectors or from the average similarity of each word to the centroid.

Although this seems like a good evaluation metric, it is important to note that TC has to be measured between two different corpora, so the topics from the training corpus can be evaluated on a corpus that has never been contacted by the model [54]. This may be an issue when trying to use datasets not already prepared for NLP tasks, especially if they do not contain any labels. A way to overcome this is to either compare it to a corpus that is known to be similar to the one the model trained on or, when the dataset used is 'general' enough, use Wikipedia's pages as a comparison, as done in Lisena et al. [2]. Nonetheless, it is important to note that if Wikipedia does not reflect the content of the corpus at hand, the results will tend to not be as positive. Reflecting this TC's characteristic, in Doogan & Buntine [55], the authors found TC to not always be a good metric to use, as although performing well in general datasets, when these are more case-specific (in their case, a Twitter dataset), that tends not to happen.

Lastly, in Dieng et al. [56], the authors proposed Topic Diversity to measure how different the top N (e.g., 25) words obtained across topics are by computing the percentage of unique words.

3- Research Methodology

All experiments were conducted in the Python programming language and the libraries employed are going to be addressed when appropriate. Figure 1 summarizes the steps followed in the methodology, which will be addressed during the present section.

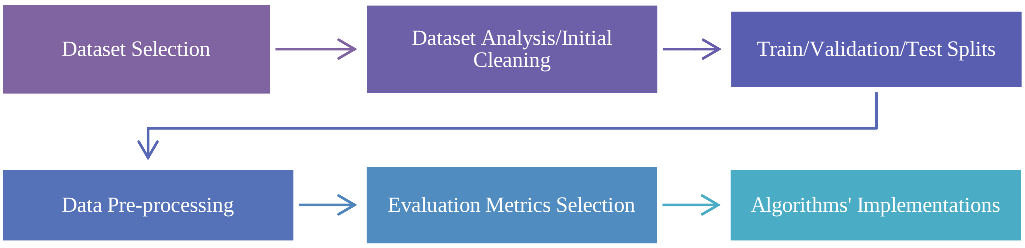
Dataset Selection
Dataset Analysis/Initial Cleaning
Train/Validation/Test Splits
Data Pre-processing
Evaluation Metrics Selection
Algorithms' Implementations

Figure 1. Methodology Steps

The datasets used throughout the experiments were chosen with two criteria in mind: being publicly available and/or being a well-known dataset in this field of research. Consequently, this work ensures it is reproducible, while providing easily comparable results.

Firstly, the 20 Newsgroup (20NG) dataset [57], available in Scikit-learn[*], includes 18 846 messages from the Usenet platform, in regards to 20 different subjects. This dataset was chosen due to its popularity in the literature for TM applications, both for new models' proposals [39, 40, 58] and for comparisons in this field [2, 8, 48, 59]. In this work, only the main text is used. Additionally, the dataset was divided into train, validation and test set, ensuring the equal division of documents according to their labels. In what concerns the cleaning of each set, the documents were subjected to a pre-processing pipeline, which will be described further ahead.

In regard to the Yahoo! Q&A dataset, it originally has 4 483 032 questions, distributed over 19 areas, and is available through the Webscope program[†]. Due to computational limitations, in our experiments we use a sample of 87 362 elements, available in the Hugging Face platform[‡], containing information regarding the question made, the answer considered the best, and the remaining. As input to the experiments, a concatenation of the question and the best answer is used. Through an analysis of the dataset, the observations corresponding to the categories 'Yahoo!7 Products' and 'Asia Pacific' were disregarded, due to corresponding to 0.05% of the dataset, not containing enough documents for the experiments to be conducted.

In what concerns the split of the dataset between train, validation, and test sets, this was done ensuring 60% of elements belonged to the train set, 20% to the validation set, and 20% to the test set. Once more, we ensured that there was an equal proportion of topics within each set. Additionally, all sets of data were subjected to a pre-processing pipeline. Lastly, regarding the Big Patent dataset [60], it originally contains approximately 1.3 million entries, with information regarding the full patent text, the respective abstract, and the label provided according to the area of study. In this work, only the abstracts are utilized, due to their smaller length. The complete original dataset, whose documents are ununiformly divided through nine subjects, is available through the Hugging Face platform[§].

[*] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

[†] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11

[‡] https://huggingface.co/datasets/yahoo_answers_qa

[§] https://huggingface.co/datasets/big_patent

The dataset is already divided into train, validation, and test sets, with the different topics equally represented across them. From this initial partition, a sample of 42 000 entries is obtained, in order to be more in line with the remaining datasets in this work: the train set contains 25 000 documents, while the validation and test sets both have 8 500 abstracts. This sample is obtained through the following steps:

- The definition of the sample size. The value of 42000 documents is arbitrarily defined, making sure that the division within the train, validation, and test sets is approximately equal to 60%, 20%, and 20%, respectively;
- The percentage of each topic per set in the original dataset is computed, so it can be reproduced in the sample;
- Considering the previous percentage, the exact number of documents to include in each set per category is found;
- A random subset is found among the original dataset's partitions to reflect the number of documents found in the previous step.

It's worth mentioning that the datasets described contain the documents' respective labels, according to the category in which they are inserted. Although these are not used during training, they are utilized to ensure that the proportion of documents belonging to a certain topic is similar in the training, validation, and test sets.

From the work conducted by Lisena et al. [2], one can infer that the algorithms are trained on a training set and then evaluated on a test set, without mention of any validation set. In the specific case of the 20NG dataset, the authors claim the model trained on the 20NG dataset was then 'tested against the same dataset'. However, from a data science perspective, it does not make sense to optimize the different models with the same test set on which the final evaluation results are obtained. This being said, the approach taken in this work's experiments will imply that the complete original dataset is divided into three partitions: training, validation, and testing. The first two will be used during the parameter tuning phase, by training the model on the training set and evaluating it on the validation one. In the final phase, these two sets will be combined and used as training sets against the test sets to find the best-performing model.

Nonetheless, it is important to note that the distinction between training, validation, and test sets is only possible given the existence of labels, due to the nature of these datasets, which are publicly available to the research community. For a task where this information is unavailable, the evaluation metrics would have to be obtained by comparing the topics obtained to another dataset. For instance, in Lisena et al. [2], the authors refer to a Wikipedia corpus to check how good a solution is. Nonetheless, even assuming Wikipedia contains information on the most diverse subjects, it may not be appropriate to use it when a dataset's nature is too specific, and the documents do not address several areas.

In what concerns the pre-processing pipeline followed in this work, it goes in line with the works of Lisena et al. [2] and Harrando et al. [8], consisting of the following steps:

- Removal of all numbers;
- Removal of all punctuation;
- Lowercasing of all documents;
- Removal of stop-words, though the Gensim[*] library;
- Lemmatization, through the Wordnet module of the NLTK[†] platform;
- Removal of all words whose length is lower than three;
- Removal of blank documents resulting from the previous steps.

It is noteworthy that despite some algorithms employed theoretically do not require pre-processing to keep consistency across results, and since none of the pre-processing steps applied change the results significantly (e.g., lemmatization is the step with a higher influence sphere, but its applicability should only slightly improve topics' readability), these were nonetheless performed.

Regarding the evaluation metrics employed in this work, our main focus is on TC. As mentioned in the Theoretical Background section, TC introduces a quantitative way of assessing the coherence of a set of topics, trying to mimic human perception in an automated process. This metric's intuition is to compare the set of words that best characterizes each topic with the documents present in a corpus in order to discover how coherent these two sets are, being that the 'coherence of a word set measures the degree that a subset is supported by another subset' [59].

The work of Röder et al. [59] introduces a four-dimension (Segmentation, Probability Calculation, Confirmation Measure and Aggregation) framework to compute TC, allowing to easily distinguish between approaches. We will use it to briefly cover the key differences between the four Coherences used in this work, which are available on Gensim[‡],

[*] https://radimrehurek.com/gensim/

[†] https://www.nltk.org/index.html

[‡] https://radimrehurek.com/gensim/models/coherencemodel.html

ensuring this work's reproducibility. It is noteworthy that all metrics took into consideration the top 10 words of each topic.

Starting by the UMASS TC [51], it can be represented by Equation 1.

$$UMASS\ Coherence = \frac{2}{N \times (N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P\ (wi,wj) + \varepsilon}{P(wj)} \tag{1}$$

where wi and wj are two different words, N equals the corpus size and ε is a parameter added to avoid the logarithm of zero, that should by default be equal to 1.

Firstly, regarding Segmentation, each word pair reflects a comparison between a word and the one which comes in the next position, given an ordered set of words that best describes each topic. In what concerns Probability Estimation, UMASS recurs to the Boolean document method to calculate the joint probability, by assessing in how many documents the word pair occurs, dividing it by the total number of elements in the corpus. Then, in the Confirmation Measure stage, the log-conditional-probability measure is used to obtain a value for each subset of word pairs, which are then (Aggregation) subject to a simple arithmetic mean to obtain a final overall TC value for the topics found by a model.

In this work, this metric is computed due to comparability reasons: whilst not being the main metric chosen to decide which parameters are best in the parameter tuning phase, it may be useful to compare across models and to be available for readers who wish to compare with the results of their endeavours.

In what concerns the UCI measure [49], based on PMI, it can be computed through Equation 2:

$$\text{UCI Coherence} = \frac{2}{N \times (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI\ (wi, wj) \tag{2}$$

where PMI (wi, wj) = $\log \frac{P\ (wi,wj) + \varepsilon}{P\ (wi)\ \times P\ (wj)}$, wi and wj are two different words, N equals the corpus size and ε is a parameter added to avoid the logarithm of zero, that should by default be equal to 1.

Contextualizing this metric in light of the structure proposed by the authors in Röder et al. [59], the Segmentation step is done by pairing each word with every other single word. After this (Probability Estimation), a Boolean sliding window of size 10 is introduced, which enables this technique to incorporate some information concerning the words' closeness. Regarding the Confirmation Measure chosen, the log-ratio (PMI) is selected, followed by the computation of the arithmetic mean to obtain the value of the coherence for all topics (Aggregation stage).

Lastly, when proposed, the researchers in Newman et al. [49] found this metric to be positively correlated with human assessment, meaning this metric has a traditional important role in evaluation. Nonetheless, as it will be shortly addressed, recent research has dethroned it as the one with the highest correlation.

Regarding the NPMI Coherence, it was proposed in Aletras & Stevenson [48] and can be computed through Equation 3:

$$\text{NPMI Coherence} = \frac{2}{N \times (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} NPMI\ (wi, wj) \tag{3}$$

where NPMI $(wi, wj) = \frac{PMI\ (wi,wj)}{-\ \log(P(wi,wj)\ +\ \varepsilon)}$; PMI (wi, wj) = $\log \frac{P\ (wi,wj) + \varepsilon}{P\ (wi)\ \times P\ (wj)}$; wi and wj are two different words, N equals the corpus size and ε is a parameter added to avoid the logarithm of zero, that should by default be equal to 1.

Decomposing the metric according to the framework proposed in Röder et al. [59], the steps obtained are exactly the same as the ones described for UCI Coherence, with the only difference being the Probability Estimation stage, where the normalized log-ratio (NPMI) measure is utilized in place of the log-ratio (PMI) measure.

In Aletras & Stevenson [48], the authors found NPMI TC to be the metric with the highest correlation with human assessments, in the detriment of the UCI and UMASS Coherences. Being this said, at that moment, this metric was considered to be the best option to be in line with human perception.

Lastly, the Context Vectors (CV) Coherence, proposed in Röder et al. [59], can be obtained through the following decomposition, per the authors' own framework: regarding Segmentation, the authors utilize the method of comparing each word to 'the total word set using words context vectors'; moreover, in what concerns Probability Estimation, a sliding window of size 110 is employed; for the Confirmation Measure stage, the cosine similarity between the NPMI of a vector and the remaining is used; lastly, the Aggregation is dealt with in line with the other metrics, through the computation of the arithmetic mean, to obtain an overall TC value, reflecting all topics.

Additionally, the authors found this metric to obtain the highest correlation with human assessment of topics, followed by NPMI Coherence, among the other metrics present in this work. Being this said, in this work, CV Coherence will be the primary metric used, followed by NPMI Coherence if needed to resolve any impasse situations.

Lastly, we will analyze the values for Topic Diversity [56], defined as the 'percentage of unique words in the top 25 words of all topics'. Despite being a rather non-complex measure, it offers some insight on a model's ability to provide distinct words to describe each topic, reflecting a model's capacity to generate topics that are able to be described by different words, which is the expected outcome of a good solution. We compute this metric by assessing how many unique words exist in the total word universe in the topics found, defining 25 as the number of words per topic that should be taken into consideration.

Regarding the algorithms' implementations, LDA's used implementation [61], available on Gensim[*], builds on the original LDA algorithm but is modified to allow for faster execution and to effortlessly deal with large sets of documents. In what concerns the NMF model, the implementation employed [62] can also be found on Gensim[†]. This variation is able to deal efficiently with large datasets, including 'sparsely corrupted data'. Top2Vec[‡], PVTM[§], and BERTopic[**] were implemented according to the respective authors' implementations.

4- Results and Discussion

To ensure fair comparisons across datasets and algorithms, regardless of the dataset at hand, the first step consisted of dividing the data into training, validation, and test sets through a valid and appropriate division process. Subsequently, parameter tuning was conducted. All five algorithms were applied to each dataset with various parameter combinations, aiming to identify the best values for each dataset-algorithm pair. To account for the variability in solutions, a random state was defined, enabling differences in results to be attributed solely to parameter changes rather than other factors.

It is important to note that the training set was exclusively used for model training, while the validation set was utilized to evaluate the results based on the chosen evaluation metrics. Additionally, as already justified in the methodology section, the primary metric employed is CV coherence, with NMPI coherence serving as an alternative if necessary. The remaining metrics were computed and presented only for potential future comparisons to ensure the comparability of results. Nonetheless, the Topic Diversity's value will be taken into consideration when analyzing a solution, as it provides information beyond the topic's coherence.

The process of parameter tuning involved a combination of empirical evidence, information from the algorithms' respective papers, default values provided by the implementations, and results obtained in other studies where comparisons were made recurring to the same dataset-algorithm pairs. Tables 2 to 6 summarize the parameters used for each algorithm according to the dataset at hand. In these tables, the best parameter values are signaled in bold, while the default parameters provided by the implementations are marked in italic style. Testing parameters from several sources allowed us to select the most suitable parameter configurations for each algorithm-dataset pair, ensuring a robust and well-informed evaluation process.

Table 2. **Parameters Evaluated for NMF Algorithm (best parameters are highlighted in bold)**

Dataset **Parameters**	**20NG**	**Yahoo! Q&A**	**BIG Patent**
Chunksize	**500**, 1000, 1500, 2000	**1500**, 2000, 2500, 3000	1500, **2000**, 2500,3000
Kappa	0.5, 0.7, **1**	0.5, 0.7, **1**	0.5, **0.7**, 1
Minimum_Probability	**0.01**, 0.05, 0.1	**0.01**, 0.05, 0.1	0.01, **0.05**, 0.1
w_max_iter	**200**, 250	**200**, 250	**200**, 250
h_max_iter	**50**, 75	**50**, 75	50, **75**
Eval_every	**10**, 20	**10**, 20	**10**, 20
Num_Topics	20	19	9

Table 3. **Parameters Evaluated for LDA Algorithm (best parameters are highlighted in bold)**

Dataset **Parameters**	**20NG**	**Yahoo! Q&A**	**BIG Patent**
Chunksize	**500**, 1000, 1500, 2000	1500, 2000, 2500, **3000**	**1500**, 2000, 2500, 3000
Alpha	Symmetric, **Auto**	Symmetric, **Auto**	Symmetric, **Auto**
Decay	**0.5**, 0.7, 0.9	**0.5**, 0.6, 0.7, 0.8, 0.9	0.5, 0.6, **0.7**, 0.8, 0.9
Offset	**1**, 4, 36, 64	1, **4**, 36, 64	1, **4**, 36, 64
Iterations	**50**, 75	**50**, 100	**50**, 100
Num_Topics	20	19	9

[*] https://radimrehurek.com/gensim/models/ldamodel.html

[†] https://radimrehurek.com/gensim/models/nmf.html

[‡] https://github.com/ddangelov/Top2Vec

[§] https://github.com/davidlenz/pvtm

[**] https://github.com/MaartenGr/BERTopic

Table 4. Parameters Evaluated for PVTM Algorithm (best parameters are highlighted in bold)

Parameters \ Dataset	20NG	Yahoo! Q&A	BIG Patent
Vector_size	**50**, 150, 200, 300	50, 150, **300**	**50**, 150, 300
Window	**1**, 10, 20	**1**, 10, 20	**1**, 10, 20
Epochs	1, **15**, 30	1, 15, **30**	1, **25**
Min_count	5, **30**, 50	5, 50, **90**	5, **30**, 50
Alpha	0.01, 0.025, **0.1**	**0.1**, 0.01	0.01, 0.025, **0.1**
Covariance_type	Diag, **Full**	Diag, **Full**	Diag, **Full**
Num_Topics	20	19	9

For the 20NG dataset, when used in conjunction with the NMF, LDA, and PVTM models, we incorporated the parameters[*] that were identified as producing the best results in the experiments conducted by Harrando et al. [8]. Moreover, default parameters were included for almost all datasets and algorithms. Regarding LDA, we referred to the findings of Hoffman et al. [61], where the authors determined the values of 0.5 and 64 to be optimal for 'Decay' and 'Offset', respectively. Hence, these values were included in our analysis. Furthermore, we introduced some additional values based on those investigated by Hoffman et al. [61] to provide a thorough exploration of parameter space.

Regarding the Top2Vec and BERTopic algorithms, the parameter tuning involved selecting values from the default values configured in their respective implementations. Additionally, we have combined these default values with similar values that make sense for each specific case. For instance, the number of documents in a dataset influences the reasonability of some parameter values, which we considered during the selection process.

Regarding the parameters for NMF, Chunksize corresponds to the number of documents to use in each training chunk; Kappa is the gradient descent step size; Minimum_Probability is used to filter out topics with smaller probabilities; w_max_iter is the maximum number of iterations to train w (e.g., word-topic matrix) per batch; h_max_iter is the maximum number of iterations to train h (e.g., topic-document matrix) per batch; eval_every is the number of batches after which the l2 norm of *v-Wh* is computed; and num_topics is the number of topics to extract from the documents.

In regard to the parameters used for LDA, Chunksize corresponds to the number of documents to use in each training chunk; Alpha refers to the default prior for document-topic distribution; Decay defines the percentage of the previous lambda value that is forgotten when assessing a new document. Offset is used to control the magnitide with which the first steps are slowed down in the first iterations; Iterations is the maximum number of iterations done on a corpus when aiming at capturing the corpus' topic distribution, and num_topics is the number of topics to extract from the documents.

The tuned parameters for the PVTM model were the vector_size, which defines the dimensionality of the feature vectors; the window, corresponding to the window size in Doc2Vec; epochs, which refer to the number of training epochs; min_count, used to define the minimal number of times a word has to appear to be considered in Doc2Vec; alpha, which corresponds to the initial learning rate; covariance_type, defying the covariance type employed during the use of the GMM model; and num_topics is the number of topics to extract from the documents.

For Top2Vec, the parameters taken into consideration were min_count, which defines the minimal frequency needed for a word to be considered; n_neighbors, which constrains the size of the local neighborhood UMAP considers when learning from the data; n_components defines the dimensionality of the reduced dimension; min_dist offers the minimum distance between points in the low-dimensional space; and min_cluster_size corresponds to the minimum number of observations needed to consider a cluster.

Table 5. Parameters Evaluated for Top2Vec Algorithm (best parameters are highlighted in bold)

Parameters \ Dataset	20NG	Yahoo! Q&A	BIG Patent
Min_count	**30**, 40, 50	**50**, 70, 90	30, 50, **70**
N_neighbors (UMAP)	**10**, 15, 20	40, **50**, 60	20, 30, 40, **50**
N_components (UMAP)	**5**, 10, 15	**5**, 10, 15	**5**, 10, 15
Min_dist (UMAP)	0, 0.1, 0.2, **0.3**	0, 0.1, **0.2**	0, 0.1, **0.2**
Min_cluster_size (HDBSCAN)	15, 30, 45, **60**	30, 60, **90**	45, 60, 75, **90**

[*] https://github.com/D2KLab/ToModAPI/blob/master/params.md

Table 6. Parameters Evaluated for BERTopic Algorithm (best parameters are highlighted in bold)

Dataset / Parameters	20NG	Yahoo! Q&A	BIG Patent
Min_topic_size	10, **20**, 30	**30**, 60, 90	20, **40**, 60
N_neighbors (UMAP)	**10**, 15, 20	**40**, 50, 60	20, **30**, 40
N_components (UMAP)	**5**, 10, 15	5, **10**, 15	**5**, 10, 15
Min_dist (UMAP)	**0**, 0.1	**0**, 0.1, 0.2	0, **0.1**, 0.2
Min_cluster_size (HDBSCAN)	15, 30, 45, **60**	30, 60, **90**	45, 60, **75**

Lastly, BERTopic was tuned considering the parameters min_topic_size, which sets the minimum number of elements a topic must have to be considered as such; n_neighbors, which constrains the size of the local neighborhood UMAP considers when learning from the data; n_components defines the dimensionality of the reduced dimension; min_dist offers the minimum distance between points in the low-dimensional space; and min_cluster_size corresponds to the minimum number of observations needed to consider a cluster.

After completing all optimizations and identifying the best parameters for each dataset-algorithm pair, the previously separate train and validation sets are merged into a new train set. All algorithms are then applied ten times to this combined train set to account for result variability. The evaluations are performed using the original test set. Since the differences across different runs of the same parameters are of interest, a random state is not defined, allowing the differences to be observable.

Tables 7 to 9 summarize the results obtained for each dataset, containing the average coherence metric and topic diversity, along with their respective standard deviations. The values marked in bold correspond to the best values achieved by each metric.

Table 7. Mean and Standard Deviation (between brackets) for all metrics and algorithms, for 20NG

Algorithm / Metric	NMF	LDA	PVTM	Top2Vec	BERTopic
CV Coherence	0.512 (0.041)	0.526 (0.012)	0.542 (0.019)	**0.703** (0.052)	0.636 (0.015)
NPMI Coherence	-0.085 (0.033)	-0.097 (0.016)	0.040 (0.016)	0.036 (0.011)	**0.056** (0.010)
UCI Coherence	-3.963 (0.641)	-4.190 (0.397)	**-0.341** (0.374)	-2.220 (0.539)	-1.324 (0.200)
UMASS Coherence	-5.671 (0.644)	-5.789 (0.597)	**-2.188** (0.224)	-4.565 (0.800)	-3.051 (0.193)
Topic Diversity	0.762 (0.033)	0.873 (0.016)	0.531 (0.014)	**0.971** (0.013)	0.847 (0.011)

Table 8. Mean and Standard Deviation (between brackets) for all metrics and algorithms, for Yahoo! Q&A

Algorithm / Metric	NMF	LDA	PVTM	Top2Vec	BERTopic
CV Coherence	0.517 (0.024)	0.476 (0.021)	0.481 (0.030)	**0.575** (0.136)	0.570 (0.033)
NPMI Coherence	0.055 (0.015)	-0.035 (0.033)	0.015 (0.033)	**0.076** (0.087)	0.063 (0.028)
UCI Coherence	**-0.088** (0.277)	-2.538 (0.751)	-0.584 (0.772)	-1.420 (1.024)	-1.832 (0.634)
UMASS Coherence	**-3.549** (0.228)	-6.202 (0.873)	-3.814 (0.808)	-5.791 (0.375)	-7.080 (0.799)
Topic Diversity	0.568 (0.014)	0.795 (0.025)	0.445 (0.021)	**0.865** (0.070)	0.762 (0.038)

Table 9. Mean and Standard Deviation (between brackets) for all metrics and algorithms, for BIG Patent

Metric \ Algorithm	NMF	LDA	PVTM	Top2Vec	BERTopic
CV Coherence	0.537 (0.014)	0.532 (0.030)	0.568 (0.012)	**0.636** (0.026)	0.561 (0.097)
NPMI Coherence	0.064 (0.007)	0.049 (0.015)	0.069 (0.004)	**0.079** (0.017)	0.070 (0.038)
UCI Coherence	0.370 (0.073)	-0.052 (0.356)	**0.423** (0.034)	-1.014 (0.432)	-0.457 (0.221)
UMASS Coherence	-2.153 (0.069)	-2.707 (0.472)	**-1.914** (0.038)	-4.605 (0.549)	-3.870 (0.346)
Topic Diversity	0.634 (0.025)	0.707 (0.031)	0.536 (0.017)	**0.981** (0.011)	0.716 (0.014)

The analysis of the experimental results across all datasets reveals that the Top2Vec algorithm consistently outperforms others when considering CV coherence. It is noteworthy that Top2Vec is a relatively recent and advanced approach to Topic Modeling, making its success in this study particularly relevant. Top2Vec is followed by either BERTopic or PVTM, depending on the specific evaluation metric. These three algorithms represent the latest advancements in TM among those under study.

The evaluation of Topic Diversity provides an insightful perspective on the algorithms' performance. Top2Vec stands out with significantly higher average values in this metric compared to all other algorithms. This finding highlights Top2Vec's ability to generate topics described by a range of distinct words.

Additionally, when comparing the three referred models (Top2Vec, PVTM, and BERTopic) in terms of Topic Diversity, more pronounced differences emerge. PVTM exhibits a weaker performance in this aspect, generating topics with lower quality regarding the distinct words provided. This limitation might impact its applicability in scenarios where topic diversity is crucial.

Contrary to the expectations based on the findings in Lisena et al. [2] and Harrando et al. [8], LDA and other classic approaches to Topic Modeling have not demonstrated superior performance compared to all other algorithms in our analysis. Instead, the results indicate that more recent techniques, such as Top2Vec and BERTopic, have outperformed these traditional methods.

This unexpected outcome suggests a shift in the landscape of Topic Modeling, where recent algorithms that leverage advanced methodologies and representations are proving to be more effective in capturing the underlying structures and semantic relationships within the data. The limitations of LDA and other classic techniques might stem from their reliance on simplistic assumptions or predefined number of topics, which may hinder their ability to handle the complexity and diversity of more complex textual data.

The prominence of Top2Vec and BERTopic in our experiments emphasizes the importance of adopting state-of-the-art approaches for Topic Modeling tasks, especially in the context of the ever-growing digital textual data and the need for more sophisticated topic representations. As we have found, the ability of Top2Vec to generate diverse and coherent topics, along with BERTopic's competitive performance, indicates that these models are better equipped to address the challenges posed by large-scale unstructured textual data.

5- Conclusion

This work aimed to address the lack of studies with recent state-of-the-art algorithms in the field of Topic Modeling. This was achieved by comparing several models across different datasets, ensuring comparability, consistency, and transparency. Additionally, only open-source datasets and algorithm implementations were used, providing a followable framework for anyone who wishes to reproduce these results or build on them for future research. Through the evaluation metric of CV coherence, Top2Vec was found to be the best-performing algorithm across all datasets, followed by BERTopic and PVTM. These three models leverage recent embedding techniques and were thus expected to outperform the more traditional algorithms under study: LDA and NMF.

Although this work attempts to provide a comparison of diverse TM algorithms, thus including representative models of different families, it is non-exhaustive, leaving potentially valid approaches unexplored. Similarly, although the datasets employed were selected to illustrate distinct contexts, including documents with different structures, there is no guarantee that slight variations in use case or context will not yield different results, including the best-performing model presented in this work.

Researchers can take advantage of this work's framework to analyze other datasets that they find relevant. Additionally, due to the continuous interest in the Text Mining field, the tendency will be for new developments in word embedding techniques and/or new algorithms to appear, which can outperform the ones employed in this work and justify a new comparative analysis, although the models in this work cover the current state-of-the-art. Lastly, the development of new evaluation metrics that claim to be able to more accurately assess the topics obtained by an algorithm may be a reason for this work to be re-assessed.

6- Declarations

6-1- Author Contributions

Conceptualization, A.A. and F.B.; methodology, A.A. and F.B.; software, A.A.; validation, F.B.; formal analysis, A.A. and F.B.; investigation, A.A.; resources, A.A.; data curation, A.A.; writing—original draft preparation, A.A.; writing—review and editing, F.B.; supervision, F.B.; project administration, F.B.; funding acquisition, F.B. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

Publicly available datasets were analyzed in this study. This data can be found here:

https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html,

https://huggingface.co/datasets/big_patent and

https://huggingface.co/datasets/yahoo_answers_qa.

6-3- Funding

This work was supported by a grant of the Portuguese Foundation for Science and Technology ("Fundação para a Ciência e a Tecnologia"), DSAIPA/DS/0116/2019, and project UIDB/04152/2020—Centro de Investigação em Gestão de Informação (MagIC).

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Li, X., & Lei, L. (2021). A bibliometric analysis of topic modeling studies (2000–2017). Journal of Information Science, 47(2), 161–175. doi:10.1177/0165551519877049.

[2] Lisena, P., Harrando, I., Kandakji, O., & Troncy, R. (2020). TOMODAPI: A Topic Modeling API to Train, Use and Compare Topic Models. Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS). doi:10.18653/v1/2020.nlposs-1.19.

[3] Hasan, Md., Hossain, Md. M., Ahmed, A., & Rahman, M. S. (2019). Topic Modeling: A Comparison of The Performance of Latent Dirichlet Allocation and LDA2vec Model on Bangla Newspaper. 2019 International Conference on Bangla Speech and Language Processing (ICBSLP). doi:10.1109/icbslp47725.2019.202047.

[4] Mohammed, S. H., & Al-Augby, S. (2020). LSA & LDA topic modeling classification: Comparison study on E-books. Indonesian Journal of Electrical Engineering and Computer Science, 19(1), 353–362. doi:10.11591/ijeecs.v19.i1.pp353-362.

[5] O'Callaghan, D., Greene, D., Carthy, J., & Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. Expert Systems with Applications, 42(13), 5645–5657. doi:10.1016/j.eswa.2015.02.055.

[6] Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020). Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. Frontiers in Artificial Intelligence, 3, 42. doi:10.3389/frai.2020.00042.

[7] Bennett, A., Misra, D., & Than, N. (2021). Have you tried Neural Topic Models? Comparative Analysis of Neural and Non-Neural Topic Models with Application to COVID-19 Twitter Data, 1-7. doi:10.48550/arXiv.2105.10165.

[8] Harrando, I., Lisena, P., & Troncy, R. (2021). Apples to Apples: A Systematic Evaluation of Topic Models. Proceedings of the Conference Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications. doi:10.26615/978-954-452-072-4_055.

[9] Calheiros, A. C., Moro, S., & Rita, P. (2017). Sentiment Classification of Consumer-Generated Online Reviews Using Topic Modeling. Journal of Hospitality Marketing & Management, 26(7), 675–693. doi:10.1080/19368623.2017.1310075.

[10] Zhou, D., Chen, L., & He, Y. (2014). A Simple Bayesian Modeling Approach to Event Extraction from Twitter. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). doi:10.3115/v1/p14-2114.

[11] Tang, H., Li, M., & Jin, B. (2019). A Topic Augmented Text Generation Model: Joint Learning of Semantics and Structural Features. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). doi:10.18653/v1/d19-1513.

[12] Haghighi, A., & Vanderwende, L. (2009). Exploring content models for multi-document summarization. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09. doi:10.3115/1620754.1620807.

[13] Boyd-Graber, J., Blei, D., & Zhu, X. (2007, June). A topic model for word sense disambiguation. Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), 28-30 June, 2007, Prague, Czech Republic.

[14] Zhao, B., & Xing, E. (2007). HM-BiTAM: Bilingual topic exploration, word alignment, and translation. Advances in Neural Information Processing Systems, 20, 3-6 December, 2007, Vancouver, Canada.

[15] Luo, W., Stenger, B., Zhao, X., & Kim, T.-K. (2015). Automatic Topic Discovery for Multi-Object Tracking. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1). doi:10.1609/aaai.v29i1.9789.

[16] Grimmer, J. (2010). A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. Political Analysis, 18(1), 1–35. doi:10.1093/pan/mpp034.

[17] Chen, Y., Ghosh, J., Bejan, C. A., Gunter, C. A., Gupta, S., Kho, A., Liebovitz, D., Sun, J., Denny, J., & Malin, B. (2015). Building bridges across electronic health record systems through inferred phenotypic topics. Journal of Biomedical Informatics, 55, 82–93. doi:10.1016/j.jbi.2015.03.011.

[18] Reisenbichler, M., & Reutterer, T. (2019). Topic modeling in marketing: recent advances and research opportunities. Journal of Business Economics, 89(3), 327–356. doi:10.1007/s11573-018-0915-7.

[19] Fino, E., Hanna-Khalil, B., & Griffiths, M. D. (2021). Exploring the public's perception of gambling addiction on Twitter during the COVID-19 pandemic: Topic modeling and sentiment analysis. Journal of Addictive Diseases, 39(4), 489–503. doi:10.1080/10550887.2021.1897064.

[20] Poongodi, M., Nguyen, T. N., Hamdi, M., & Cengiz, K. (2021). Global cryptocurrency trend prediction using social media. Information Processing & Management, 58(6). doi:10.1016/j.ipm.2021.102708.

[21] Saura, J. R., Ribeiro-Soriano, D., & Zegarra Saldaña, P. (2022). Exploring the challenges of remote work on Twitter users' sentiments: From digital technology development to a post-pandemic era. Journal of Business Research, 142, 242–254. doi:10.1016/j.jbusres.2021.12.052.

[22] Saura, J. R., Palacios-Marqués, D., & Ribeiro-Soriano, D. (2023). Exploring the boundaries of open innovation: Evidence from social media mining. Technovation, 119, 102447. doi:10.1016/j.technovation.2021.102447.

[23] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391–407. doi:10.1002/(sici)1097-4571(199009)41:6<391::aid-asi1>3.0.co;2-9.

[24] Hofmann, T. (1999). Probabilistic latent semantic indexing. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. doi:10.1145/312624.312649.

[25] Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics, 5(2), 111–126. doi:10.1002/env.3170050203.

[26] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

[27] Harris, Z. S. (1954). Distributional Structure. 10(2–3), 146–162. doi:10.1080/00437956.1954.11659520.

[28] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26, 5-10 December, 2013, Lake Tahoe, United States.

[29] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. Proceedings of the 31st International Conference on Machine Learning, 21-26 June, 2014, Beijing, China.

[30] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543. doi:10.3115/v1/d14-1162.

[31] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. doi:10.18653/v1/e17-2068.

[32] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135–146. doi:10.1162/tacl_a_00051.

[33] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. doi:10.48550/arXiv.1810.04805.

[34] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3982-3992. doi:10.18653/v1/d19-1410.

[35] Yonghui Wu, Yuxin Ding, Wang, X., & Jun Xu. (2010). A comparative study of topic models for topic clustering of Chinese web news. 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China. doi:10.1109/iccsit.2010.5564723.

[36] Pietsch, A.-S., & Lessmann, S. (2018). Topic modeling for analyzing open-ended survey responses. Journal of Business Analytics, 1(2), 93–116. doi:10.1080/2573234x.2019.1590131.

[37] Wang, C., Paisley, J., & Blei, D. M. (2011). Online variational inference for the hierarchical Dirichlet process. Proceedings of the fourteenth international conference on artificial intelligence and statistics, 11-13 April, 2011, Fort Lauderdale, United Sates.

[38] Lenz, D., & Winker, P. (2020). Measuring the diffusion of innovations with paragraph vector topic models. PLoS ONE, 15(1), 0226685. doi:10.1371/journal.pone.0226685.

[39] Angelov, D. (2020). Top2vec: Distributed representations of topics. arXiv preprint. doi:10.48550/arXiv.2008.09470.

[40] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint, arXiv:2203.05794. doi:10.48550/arXiv.2203.05794.

[41] Hong, L., & Davison, B. D. (2010). Empirical study of topic modeling in Twitter. Proceedings of the First Workshop on Social Media Analytics, 80-88. doi:10.1145/1964858.1964870.

[42] Bellman, R., & Kalaba, R. E. (1965). Dynamic programming and modern control theory. Academic Press, New York, United States.

[43] McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software, 3(29), 861. doi:10.21105/joss.00861.

[44] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11), 2579-2605.

[45] Campello, R.J.G.B., Moulavi, D., Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. Advances in Knowledge Discovery and Data Mining. PAKDD 2013, Lecture Notes in Computer Science, 7819, Springer, Berlin, Germany. doi:10.1007/978-3-642-37456-2_14.

[46] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1), 11–21. doi:10.1108/eb026526.

[47] Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. doi:10.1145/290941.291025.

[48] Aletras, N., & Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. Proceedings of the 10th international conference on computational semantics (IWCS 2013), 19-22 March, Potsdam, Germany.

[49] Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics, 2-4 June, 2010, Los Angeles, United States.

[50] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., & Blei, D. (2009). Reading tea leaves: How humans interpret topic models. Advances in neural information processing systems, 22, 7-10 December, 2009, Vancouver, Canada.

[51] Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. Proceedings of the 2011 conference on empirical methods in natural language processing, 27-31 July, Edinburgh, Scotland.

[52] Lau, J. H., Newman, D., & Baldwin, T. (2014). Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. doi:10.3115/v1/e14-1056.

[53] Bouma, G. (2009). Normalized (pointwise) Mutual Information in Collocation Extraction. Proceedings of GSCL, 30, 31-40.

[54] Newman, D., Bonilla, E. V., & Buntine, W. (2011). Improving topic coherence with regularized topic models. Advances in neural information processing systems, 24, 12-14 December, 2011, Granada, Spain.

[55] Doogan, C., & Buntine, W. (2021). Topic Model or Topic Twaddle? Re-evaluating Semantic Interpretability Measures. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. doi:10.18653/v1/2021.naacl-main.300.

[56] Dieng, A. B., Ruiz, F. J. R., & Blei, D. M. (2020). Topic modeling in embedding spaces. Transactions of the Association for Computational Linguistics, 8, 439–453. doi:10.1162/tacl_a_00325.

[57] Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. Machine Learning Proceedings 1995, 331–339, Morgan Kaufmann, Burlington, United States. doi:10.1016/b978-1-55860-377-6.50048-7.

[58] Zhao, H., Du, L., Buntine, W., & Liu, G. (2017). MetaLDA: A Topic Model that Efficiently Incorporates Meta Information. 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, United States. doi:10.1109/icdm.2017.73.

[59] Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 399-408. doi:10.1145/2684822.2685324.

[60] Sharma, E., Li, C., & Wang, L. (2019). BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2204–2213. doi:10.18653/v1/p19-1212.

[61] Hoffman, M., Bach, F., & Blei, D. (2010). Online learning for latent dirichlet allocation. Advances in Neural Information Processing Systems 23 (NIPS 2010), 6-9 December, 2010, Vancouver, Canada.

[62] Zhao, R., & Tan, V. Y. F. (2017). Online Nonnegative Matrix Factorization with Outliers. IEEE Transactions on Signal Processing, 65(3), 555–570. doi:10.1109/TSP.2016.2620967.