*Computers, Materials & Continua*
DOI: 10.32604/cmc.2023.032864
***Article***



# Automated File Labeling for Heterogeneous Files Organization Using Machine Learning

**Sagheer Abbas[1], Syed Ali Raza[1,2], M. A. Khan[3], Muhammad Adnan Khan[4,*], Atta-ur-Rahman[5], Kiran Sultan[6] and Amir Mosavi[7,8,9]**

[1]School of Computer Science, National College of Business Administration & Economics, Lahore, 54000, Pakistan
[2]Department of Computer Science, GC University Lahore, Pakistan
[3]Riphah School of Computing & Innovation, Faculty of Computing, Riphah International University, Lahore Campus, Lahore, 54000, Pakistan
[4]Department of Software, Pattern Recognition and Machine Learning Lab, Gachon University, Seongnam, 13120, Korea
[5]Department of Computer Science, College of Computer Science and Information Technology (CCSIT), Imam Abdulrahman Bin Faisal University (IAU), P.O. Box 1982, Dammam, 31441, Saudi Arabia
[6]Department of CIT, The Applied College, King Abdulaziz University, Jeddah, 31261, Saudi Arabia
[7]John von Neumann Faculty of Informatics, Obuda University, Budapest, 1034, Hungary
[8]Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava, Bratislava, 81107, Slovakia
[9]Faculty of Civil Engineering, TU-Dresden, Dresden, 01062, Germany
*Corresponding Author: Muhammad Adnan Khan. Email: adnan@gachon.ac.kr
Received: 31 May 2022; Accepted: 05 July 2022

**Abstract:** File labeling techniques have a long history in analyzing the anthological trends in computational linguistics. The situation becomes worse in the case of files downloaded into systems from the Internet. Currently, most users either have to change file names manually or leave a meaningless name of the files, which increases the time to search required files and results in redundancy and duplications of user files. Currently, no significant work is done on automated file labeling during the organization of heterogeneous user files. A few attempts have been made in topic modeling. However, one major drawback of current topic modeling approaches is better results. They rely on specific language types and domain similarity of the data. In this research, machine learning approaches have been employed to analyze and extract the information from heterogeneous corpus. A different file labeling technique has also been used to get the meaningful and `cohesive topic of the files. The results show that the proposed methodology can generate relevant and context-sensitive names for heterogeneous data files and provide additional insight into automated file labeling in operating systems.

**Keywords:** Automated file labeling; file organization; machine learning; topic modeling

## 1 Introduction

Whether physical or digital, data organization is a critical part of our routine matters. The growing amount of data in modern working environments demands an effective system to manage files and folders. The file organization and management have several aspects, including labeling and naming digital files. Names and labels play a significant part in organizing the digital files stored in thousands inside computer systems. The number of digital files stored in any computer system is aggregating rapidly with the progress in the development of mass secondary storage devices [1]. With this increasing number of files, most users find it difficult to search and access any specific file. Computer users usually spend much time every day interacting with digital files and folders stored in their systems. These interactions consist of several actions, including creating, downloading, labeling, reviewing, navigating, searching for, moving, saving, copying, sharing, and deleting digital files. In addition, files that contain heterogeneous data, such as numbers, images, or sounds, are becoming popular. Users usually tend to demonstrate substantial creativity in file labeling [2]. However, the file labeling patterns are recognizable such as files being named to display the file they represent, their purpose, or a relevant creation date or deadline [3], but may also contain characters to expedite the sorting of the files to reduce clutter. Commonly, it is believed that a balanced and expressive file name may oblige multiple tenacities, including labeling, information organization, decreased searching time, and increased readability [4].

Filename generation is typically an unsupervised machine learning technique through which the hidden semantic information can be extracted from the corpus [5]. The corpus used in file labeling can typically be made up of thousands of files in the form of a data set, CSV file, and text file. Another essential aspect of this is that sometimes the user files require more than one topic or include many words. No hard and fast rule can be applied to get highly accurate and meaningful data in file labeling [6]. Therefore, file labeling relies on linguistic analysis and preprocessing techniques to filter the data to get reasonably meaningful results [1].

One of the significant issues in file labeling is analyzing the type of content and language used in the file. The data provided for file labeling first needs to go through a rigorous preprocessing stage to remove insignificant content that does not play any role in file labeling. Once effective content is extracted, the hidden feature is analyzed to compute the preprocessed file's dimensionality, structure, and size. The semantic information obtained can be further converted into one hot vector for the unique representation of each word in files because files may show many Labels, and these Labels can also overlap with each other. File labeling is also used to break down the file into different Labels based on the probability distribution. In this research, different techniques have been used, such as Latent Dirichlet Allocation (LDA) [7], Latent Semantic Analysis (LSA) [8], and Non-negative Matrix Factorization (NMF) [9] for automated file labeling. This research aims to incorporate computational linguistic analysis based on machine learning techniques. It is hypothesized that an automated system can be developed to generate contextually meaningful labels for user files to assist users in the overall management and organization of digital files inside any computer system.

## 2 Related Work

Previously, several attempts have been made in topic modeling and labeling short text and articles. Existing topic modeling tasks are mainly done using specific datasets such as fake news, ABC news dataset, New York times dataset, Twitter dataset, etc. Considering the monotony in these datasets, existing labeling and topic modeling approaches perform very well [10]. Daniel et al. [11] studied biological science-related data and analyzed that standard preprocessing approaches are not good

enough to apprehend such diversified documents. They [11] developed a standalone toolbox and extracted common daily life words through graphical representation and heat-map and graphical representation based on topic similarity. Rubayyi et al. [12] used different techniques such as LDA, LSA, and Probabilistic Latent Semantic Analysis (PLSA). They identified probability-based topics over a dataset of hundred documents based on frequent keywords.

Pantel et al. [13] proposed using a lexical-semantic pattern for labeling semantic classes, encompassing all class members to learn a different label. A significant limitation of this task was that it only worked well over the semantically homogeneous, fine-grained clusters. Alokaili et al. [14] proposed a neural approach using the sequence-to-sequence technique for document labeling that does not suffer from this limitation. The model was trained using distant supervision over a sizeable synthetic dataset. Human experts evaluated the model by comparing labels to ones generated by the proposed technique. Initially, Seung et al. [9] conducted [15] human scoring of topics. Humans evaluated topics used in a novel and directly scored different topics learned by a topic model based upon pointwise mutual information. Qiaozhu et al. [16] used unsupervised machine learning approaches for automatically labeling topics. They generated expected labels through bigrams and noun chunks and afterward ranked those expected labels based on divergence with the preselected topic. Another approach that many researchers have used is to match topic words to concepts based on knowledge base [17,18]. Jey et al. [19] extracted top N terms to select topics, while some others have used summarization approaches to create labels for topics [20,21].

One common drawback of existing topic modeling approaches is the absence of interpretable topic space [20]. LDA [7,22] and PLSA [8] are traditional topic modeling techniques and envision different documents as a concoction of some discretized topics on some fixed parameter. Angelov [23] presented the Top2Vec approach based on Word2Vec and Doc2Vec models to build a document and topic vector and an interpretable word space. Another similar approach is BERTopic [24] which separates the embedding stage from the topic creation stage in contrast to the Top2Vec approach. Top2Vec considers words adjacent to the centroid of a cluster and creates coherent and interpretable topic representations very nicely.

On the other hand, BERTopic [24] focuses on the cluster and attempts to model the topic representation from the entire cluster. This allows the topic representations to be a bit more diverse and disregards the notion of centroids. One major limitation of document embedding techniques based on training is that they do not ensure quality on heterogeneous datasets. The model can only learn semantic associations like notions and thoughts between words and sentences but cannot comprehend the central idea of the document [25].

## 3 Materials and Methods

The proposed model consists of the following modules: preprocessing the data, Document matrix (DM) and Term analysis module (TAM), Topic modeling module (TMM), and File name generation module (FNGM) as shown in Fig. 1. The dataset used in this research is an amalgam of ABC news data set, different research articles, and books. The reason for collecting various datasets is to increase heterogeneity in the data used for training the system.

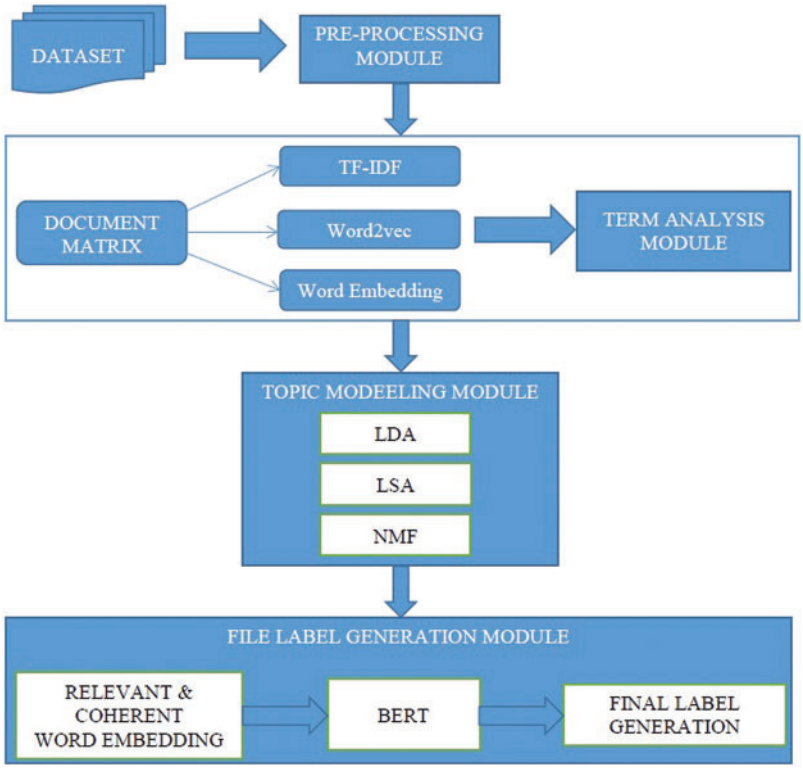



**Figure 1:** Proposed model for automated labeling of heterogeneous files

### *3.1 Preprocessing Module*

Humans have the senses to understand rich knowledge about the language, the hidden meanings, and the facts behind words used in languages. On the contrary, operating systems do not possess any such information. Most of the files in the computer system are usually classified as unstructured documents; therefore, almost all of the files must be preprocessed before file labeling begins. The preprocessing of the files includes multiple steps such as Filtering, Stop Word Removal, Tokenization, and Stemming of document words as shown in Fig. 2. This module's main idea is to convert digital files into strings. Therefore a pipeline process is required that can make sentences like "The man stands in front of the dog" and turn them into a string of words: "the," "man," "stand," "in," and "front," "of," "dog."Afterward, these words are translated into a numeric representation to make them machine-readable. Once the stop words like in, of, the, etc. are removed, the next step is to tokenize these sentences into chunks of individual words. After tokenization, the chunks of words are transformed into their stem to reduce each word to its basic form, such as energies, and energy is changed to energy. The last preprocessing step is to recognize named entities in the text. Named entity recognition categorizes the text with tags of relevant named entities such as people, location, association, phone numbers, email, etc. This module helps identify important names used as possible file labels.

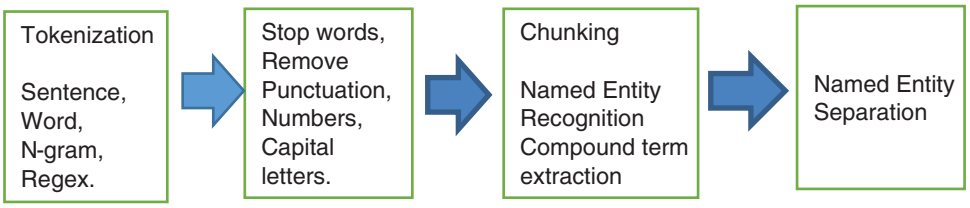



**Figure 2:** Preprocessing pipeline for proposed automated file labelling of heterogeneous files

### *3.2 Document Matrix and Term Analysis (DM-TA) Module*

The DM-TA module consists of different submodules. The purpose of these sub-modules is to represent each document better. For this purpose, Term frequency-inverse document frequency (TF-IDF), Word Embeddings, and Word2Doc Models have been used.

#### *3.2.1 TF-IDF*

Term Frequency over inverse document frequency [26] is a typical statistical method used by Natural language processing experts. In standard systems to convert text documents into the matrix representations of the feature vectors. TF-IDF scores represent the value or the relevance of the respective terms in the given set of documents. TF-IDF is computationally dependent on the vocabulary set and thus fails in the environment where there is a constant and frequent change in the text corpora.

#### *3.2.2 Word Embedding*

Word Embeddings represent a word in a high dimensional context. It may appear in a vocabulary employing a real value vector. Since word embedding preserves the real contextual purpose of the word it is used for, it yields better results in several tasks, including but not limited to similarity analysis and label extraction.

#### *3.2.3 Word2Vec*

Word2Vec [27] uses an external neural network trained on an extensive data set and represents a word as a vector in vector space where its locations represent the meaning it may be perceived. Word closers and clustering tend to have similar syntactic and semantic meanings. Word2Vec works excellently to predict the similarity of two words in their syntactical and semantic capacity but still cannot predict words in their contextual space.

#### *3.2.4 Term Analysis*

Once significant tokens are extracted through TF-IDF, word embedding, and the Word2Vec approach, these tokens are represented through Doc2Vec for term analysis. Doc2Vec is an interactive and user-friendly package to learn the vector representation of words described in [11] in Word2Vec. Each of these vectors is stored as columns of a matrix W. The sum of the ordered words in vocabulary is used as a feature to predict the next word as a classifier. The goal is to maximize average log probability. The prediction is made using softmax for a multiclass classifier where each of y(i) is a non-normalized log probability for each output word i computed using Eq. (1)

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p\left(w_t | w_{t-k}, \ldots, w_{t+k}\right) \tag{1}$$

where the log probability p is calculated in Eq. (2)

$$p\left(w_t|w_{t-k},\ldots,w_{t+k}\right)=\frac{e^{yw_t}}{\sum_i e^{y_i}} \tag{2}$$

Finally, the y is computed as a sum of b and Uh where b & U are softmax parameters and h is the resultant of the average or concatenation of the word vectors as mentioned in Eq. (3).

$$y=b+Uh\left(w_{t-k},\ldots,w_{t+k};W\right) \tag{3}$$

The neural language model is trained to predict word vectors. After the training, the words in the vector space having similar meanings are closer to each other than the contextually far apart words. These vectors can be used as input to an unsupervised clustering algorithm, among others, to assign them to individual groups based on the measure of their similarity as represented by various factors in their vector positioning in the Euclidean space.

### *3.3 Topic Modeling*

Once significant terms are identified and represented through Doc2Vec [28], the next step is to summarize these variant terms into a simple, descriptive name for a given file. After analyzing the most common topic words, this process is conducted by most experts. However, recent attempts have been made to integrate supervised mechanisms so that labels can be determined in advance to match with learned labels [29]. In this research, supervision cannot be incorporated as the aim is to develop a fully functional automated approach that can assign names to user files independently. For this purpose, LDA [7], NMF [9], and LSA [30] techniques are applied to model labels for any user file.

#### *3.3.1 LDA*

The LDA approach can represent documents as random mixtures over hidden topics, where every topic can be categorized by a distribution of words, as shown in Fig. 3.

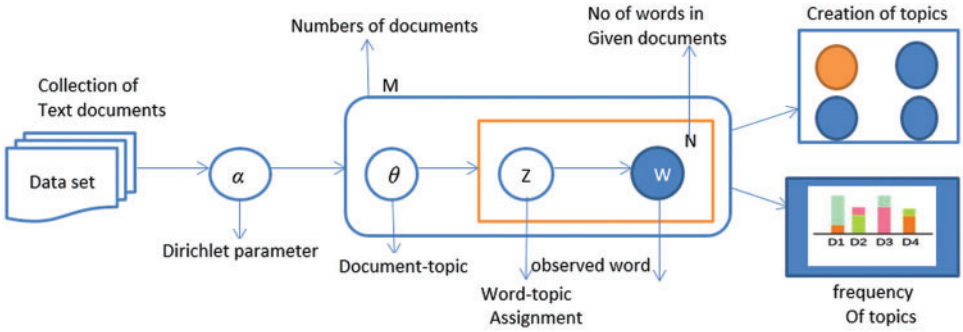



**Figure 3:** LDA model for topic modeling

The probability of a corpus (F) as mentioned in Eq. (4) [27].

$$P\left(F|\alpha,\beta\right)=\prod_{k=1}^{K}\int P\left(\omega_k|\alpha\right)\left(\prod_{l=1}^{L_k}\sum_{L_{kl}}P\left(L_{kl}|\omega_k\right)P\left(W_{kl}|L_{kl},\beta\right)\right)k\omega_k \tag{4}$$

where $\alpha$ and $\beta$ are the parameters of the Dirichlet before the per-file label and the per-label word distribution, respectively, the $\omega$ represents each word, and L represents labels. At the same time, k and l are the range of words and labels. The merging distribution is acquired for a single file F

by integrating $\omega$ with the sum of labels L. According to the Bayes theorem, if the likelihood is a multinomial distribution of labels L and weight W ($L_{kl}$, $W_{kl}$) and the prior probability is Dirichlet distributed over $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the posterior Dirichlet distribution can be easily obtained.

### *3.3.2 NMF*

One common problem with textual analysis is the curse of dimensionality. Considering many files with thousands of words having dozens of significant tokens can lead to a very complex situation and require unique mechanisms to deal with these many dimensions. NMF is one such approach that can reduce the dimensions of the data. NMF offers relatively more minor weightage to the tokens with a smaller coherence based on the factor analysis method. Suppose we have an input matrix F of k × l dimension, and we factorize this matrix F into two M & N matrices with dimension of k × v and l × w respectively. Here each column of M characterizes the weightage of every word in a sentence, while each row of N represents the word embeddings. Nevertheless, it is considered that the entries of M and N are positive. The matrices M and N will be calculated and updated iteratively until convergence over the objective function as calculated in Eq. (5) [9].

$$\frac{1}{2}||F - MN||_A^2 = \sum_{i=1}^{l}\sum_{j=1}^{k}\left(F_{ij} - (MN)_{ij}\right)^2 \tag{5}$$

The rules for updating M and N can be derived using the objective functions calculated in Eqs. (6) and (7) [9].

$$M_{ic} \leftarrow M_{ic}\frac{(FN)_{ic}}{(MMN)_{ic}} \tag{6}$$

$$N_{cj} \leftarrow N_{cj}\frac{(MF)_{cj}}{(MMN)_{cj}} \tag{7}$$

### *3.3.3 LSA*

The basic theme behind LSA approach is to break down the matrix of files into two standalone matrices, namely File label matrix (FLM) and Label term matrix (LTM). Assumed that we have k number of files and l number of tokens, we created the k × l matrix where every row characterizes a file, and every column characterizes a term. To form LTM, LSA counts the number of times any word appeared in any file using TFIDF. One issue with the LTM is that this matrix is always scarce, noisy, and redundant across almost all dimensions. Therefore we need to apply dimensionality reduction on LTM to catch some latent labels that can characterize the connotation between terms and files. Truncated Singular Value Decomposition (TSVD) [31] is applied to reduce dimensionality. TSVD is a popular approach to factorize any matrix into the product of three different matrices such that A = S × M × N where S contains the singular values of A and M, N are the factors of A. To reduce the dimensions using TSVD, we need the value of t, a hyperparameter that we can set independently. The probabilistic value of A can be estimated as mentioned in Eq. (8) [30].

$$A \approx M_t S_t N_t^T \tag{8}$$

A single label yields a single word, while different words must be incorporated to generate multiple labels. Therefore each file is reduced to a probability distribution over a set of labels. LDA defines the

joint probability P of any file F with a word W as mentioned in Eqs. (9) and (10) [31].

$$P(F, W) = P(D) \sum_z P(z|F)\, P(W|F) \tag{9}$$

$$P(F, W) = \sum_z P(z)\, P(F|z)\, P(W|z) \tag{10}$$

where P (F, W) defines the multinomial distributions that may be trained to deduce parameter estimates that are dependent on overlooked.

### *3.4 Label Generation*

Once LDA, NFM, and LSA were applied, and files were transformed into n-dimension file embeddings, these embeddings were clustered into file labels. As these embeddings were large, BERT was applied on each label cluster to extract significant terms for the final label of each file. This step is crucial as topic modeling approaches are based on statistical methods, and there is a high probability that these approaches can extract non-significant words from the files. Each extracted label represents the actual file, but the problem with these labels is that most have different and distinct words. BERT is applied to rewrite these labels into more meaningful file names to generate a more meaningful label. BERT is a pre-trained deep neural network that uses multiple transformer layers and generates a language model effectively. Another significant aspect of BERT is that it can extract multiple word embeddings from a file based on its context. In this research, BERT is used as a sentence transformer. The results generated from LDA, NFM, and LSA had different, distinct words mentioned in Fig. 4 and required transformation into a helpful file label.

| | |
|---|---|
| Topic #0: | column worthwhile prefers parallelization robustness |
| Topic #1: | gas european publicity 40 claude |
| Topic #2: | blockchain captioning ciao removes fails |
| Topic #3: | page audience capstone hypernym actions |
| Topic #4: | trillions vendor odd accidental uncontroversial |
| Topic #5: | consumption conclude kane struggles enthusiastically |
| Topic #6: | chico simmer obsession wasserstein frank |
| Topic #7: | classed arrived unetbootin practical drastically |
| Topic #8: | app hotdogs hotdog ios neural |
| Topic #9: | beach pixel fortunes undetermined mushrooms |
| Topic #10: | asian amniocentesis ᄒᆞᄂᆞᄀᆞᄀᆞᄀᆞᄋᆞᄀᆡ pandas scaled |
| Topic #11: | network model mario neural image |
| Topic #12: | Save droid pi crypto wish raspberry |
| Topic #13: | advance exterior gg plot watching ingredient |
| Topic #14: | wavenet www outside circumstances talking |
| Topic #15: | boating object annotation transcriptions determines encountered |
| Topic #16: | short drugs importantly insight american |
| Topic #17: | theodore motivation month jonze posters |
| Topic #18: | aircraft spoofing ideas felt horrible |
| Topic #19: | company unstructured reverting degree percentage |
| Topic #20: | evidence white wedding accurately corrupted |

**Figure 4:** Distinct labels generated based on topic modeling module of proposed system

## 4 Results and Discussion

Automated file labeling is an exciting and challenging task simultaneously, as it requires dealing with heterogeneous data with varying amounts of words and tokens in different files. After the preprocessing step, the number of most frequent words was observed, as shown in Fig. 5.

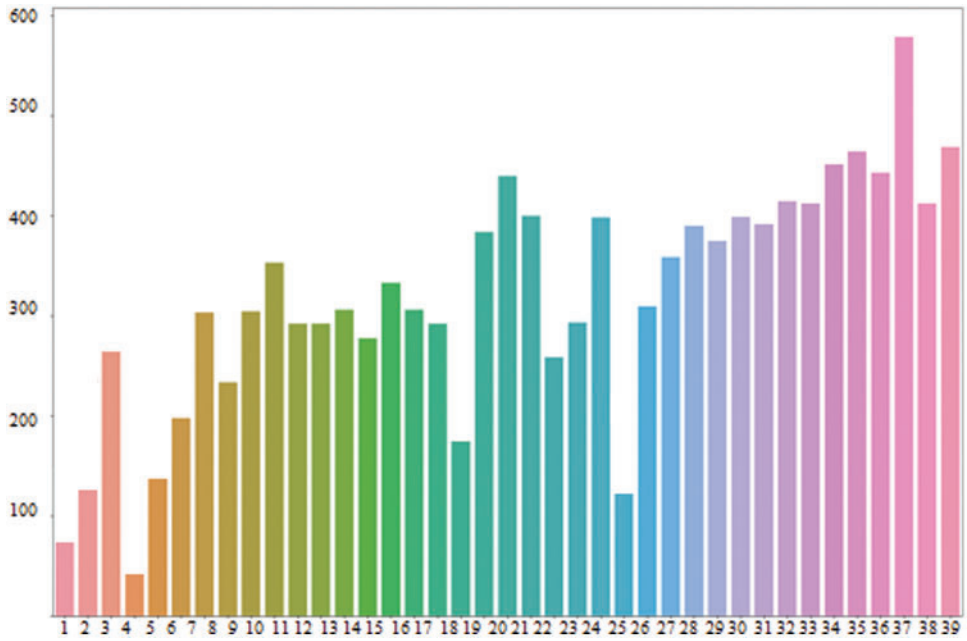



**Figure 5:** Word frequencies after preprocessing step of proposed system for automated file labeling

Based on this frequency of words, the TF-IDF extracted terms from files are mentioned in Fig. 6. Unfortunately, the TF-IDF and other document vector results were sparse, noisy, and redundant. Therefore, once the vector representation of the significant and frequent terms was completed, different unsupervised machine learning techniques were applied for label modeling. The main objective of the topic modeling module is to find a concealed theme that governs the semantics of a file, as abstract labels characterize these themes. However, realistically these results are not based on any probabilistic model, as mentioned in Tab. 1.

{'secretary_of_state': 0, 'visiting': 1, 'argentina': 2, 'latin': 3, 'american': 4, 'tour': 5, 'pledges': 6, 'united_sta
tes': 7, 'government': 8, 'documents': 9, 'light': 10, 'hundreds': 11, 'children': 12, 'captured': 13, 'imprisoned': 14,
'argentine': 15, 'military': 16, 'pres': 17, 'fernando_de_la': 18, 'rua': 19, 'recent': 20, 'political': 21, 'economic':
22, 'amount': 23, 'takes': 24, 'carlos': 25, 'president': 26, 'administration': 27, 'fight': 28, 'insists': 29, 'cabine
t': 30, 'policies': 31, 'british': 32, 'telecommunications': 33, 'plans': 34, 'invest': 35, 'billions_of_dollars': 36,
'region': 37, 'mexico': 38, 'brazil': 39, 'key': 40, 'growth': 41, 'markets': 42, 'cellular': 43, 'telephone': 44, 'mark
et': 45, 'photo': 46, 'shareholders': 47, 'rescue': 48, 'package': 49, 'agreement': 50, 'cash': 51, 'governments': 52,
'spain': 53, 'spanish': 54, 'state': 55, 'holding': 56, 'company': 57, 'owns': 58, 'percent': 59, 'airline': 60, 'provid
e': 61, 'international': 62, 'trade': 63, 'commission': 64, 'steel': 65, 'products': 66, 'japan': 67, 'countries': 68,
'ruling': 69, 'producers': 70, 'workers': 71, 'dept': 72, 'planned': 73, 'millions_of_dollars': 74, 'wide': 75, 'decide
d': 76, 'vote': 77, 'imports': 78, 'russia': 79, 'south_africa': 80, 'thailand': 81, 'industry': 82, 'cases': 83, 'agenc
y': 84, 'domestic': 85, 'companies': 86, 'similar': 87, 'federal': 88, 'court': 89, 'home': 90, 'singer': 91, 'buenos_ai
res': 92, 'turned': 93, 'museum': 94, 'show': 95, 'return': 96, 'week': 97, 'run': 98, 'public': 99, 'theater': 100, 'fo
cus': 101, 'music': 102, 'era': 103, 'display': 104, 'paper': 105, 'national': 106, 'gallery': 107, 'column': 108, 'vie
w': 109, 'regional': 110, 'world': 111, 'economy': 112, 'graph': 113, 'percentage': 114, 'change': 115, 'gross_domesti
c': 116, 'product': 117, 'controlled': 118, 'senate': 119, 'gives': 120, 'needed': 121, 'lift': 122, 'labor': 123, 'bil
l': 124, 'intended': 125, 'lower': 126, 'unemployment': 127, 'rate': 128, 'costs': 129, 'businesses': 130, 'power': 131,
'big': 132, 'weeks': 133, 'union': 134, 'new': 135, 'increasingly': 136, 'ministers': 137, 'publicly': 138, 'themselve

**Figure 6:** The most frequent terms extracted using TFIDF

**Table 1:** File labels extracted using LDA

| | Label 1 | Label 2 | Label 3 | Label 4 | Label 5 | Label 6 | Label 7 | Label 8 | Label 9 | Label 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | process | handle | would | train | fact | problem | Area | python | perform | represent |
| 2 | interest | comfortable | contact | done | would | solution | Whose | Big | logic | present |
| 3 | inspired | even | Fail | accuracy | demand | entire | Supply | Data | selection | Way |
| 4 | randomly | way | Door | imagine | intuitive | notice | Rough | university | formula | physical |
| 5 | weird | willing | One | guess | evidence | Ie | institute | center | walking | addition |

The same information is also displayed through a histogram, as shown in Fig. 7. The labels obtained through NMF are mentioned in Tab. 2. It is observable that NMF label results are not appropriate as, first of all, it repeats most words, and above all, the words extracted by NMF are not much meaningful, neither have they depicted the actual label of the file. For instance, if we consider labels 2 and 3 from Tab. 2, both contain almost similar terms as extracted by NMF, while in Tab. 1, LDA extracted utterly different words for these files.

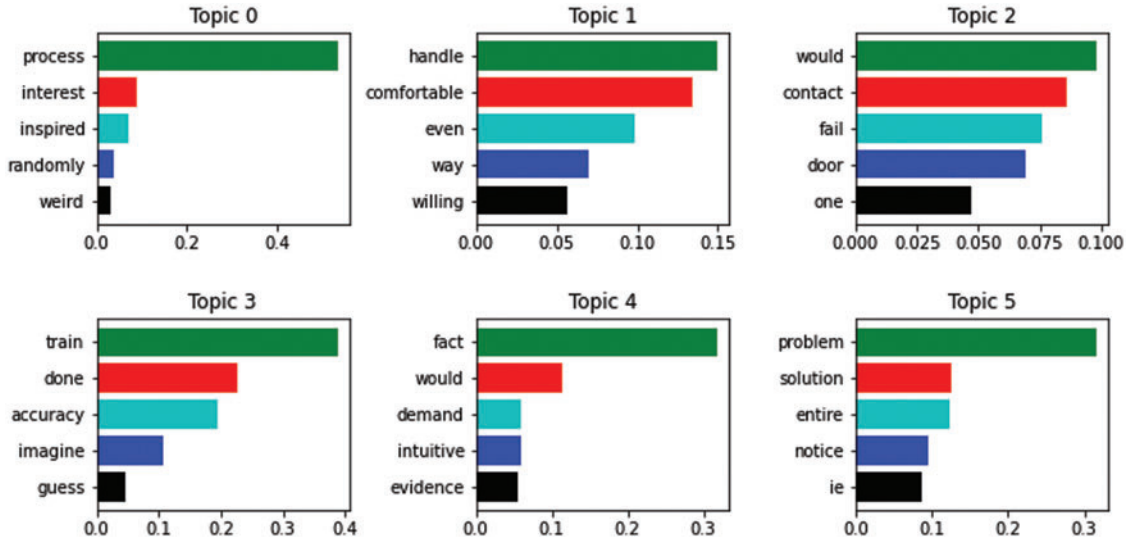



**Figure 7:** Histogram of extracted labels using LDA

**Table 2:** File labels extracted using NMF

| | Label 1 | Label 2 | Label 3 | Label 4 | Label 5 | Label 6 | Label 7 | Label 8 | Label 9 | Label 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | learning | data | one | network | machine | like | neural | Time | use | would |
| 2 | Zero | zero | zero | usually | natural | price | several | attention | zero | zero |
| 3 | Fake | fall | fall | cool | attention | zero | advantage | complete | fairly | fake |
| 4 | figure | figure | figure | zero | difference | feature | standard | writing | field | figure |
| 5 | Field | field | field | false | recognize | false | zero | feature | felt | field |

LSA is also considered one of the most fundamental approaches in label modeling. After LDA and NMF, we applied LSA to extract label words from the given corpus mentioned in Tab. 3. LSA results are much better than NMF, but LDA still got better results than LSA and NMF.

**Table 3:** File labels extracted using LSA

| | Label 1 | Label 2 | Label 3 | Label 4 | Label 5 | Label 6 | Label 7 | Label 8 | Label 9 | Label 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Learning | data | fall | network | Machine | one | neural | use | like | represent |
| 2 | One | handle | contact | machine | Neural | big | model | would | set | present |
| 3 | Go | field | field | layer | recognize | ovation | show | neural | ai | fake |
| 4 | problem | flow | door | next | Use | level | set | complete | problem | field |
| 5 | process | figure | one | problem | Output | case | way | feature | process | addition |

It is clear from Tabs. 1–3 that each LDA, NMF, and LSA extracted different terms for file labeling. Although NMF did not perform well for most of the files in some cases, such as labels 5 and 7, it extracted more meaningful terms. Furthermore, it was also observed that all of these three topic modeling approaches identified some non-significant words such as "one," "go," "like," etc. Fig. 9 depicts the inter-topic distance map for labels of each file extracted using LDA, NMF, and LSA.

The dominant labels obtained through LDA and the coherence score of these labels are shown in Fig. 8.

| Topic_Perc_Contrib | Keywords |
|---|---|
| 0.4382 | likely, hopefully, keeping, letter, example, construction, first, correction, tip, track |
| 0.4290 | improve, style, quality, combination, engagement, aware, accordingly, robustness, active, apache |
| 0.8828 | discover, independently, walking, notion, desirable, forever, tall, infinitely, extreme, unlimited |
| 0.8143 | price, house, expect, catch, example, one, would, many, time, preparation |
| 0.5399 | context, sample, expert, perfectly, time, living, make, also, consuming, bug |
| 0.4016 | problem, performance, solve, happening, use, ensemble, flexibility, anyway, strictly, wildly |
| 0.7572 | tool, nothing, plus, heart, deeply, summary, regardless, primary, ranging, many |
| 0.5016 | could, store, would, indeed, unknown, well, similar, people, also, stated |
| 0.6287 | solution, might, like, helping, way, want, u, feasible, resource, time |
| 0.6114 | available, per, free, university, week, statistical, launch, certificate, eight, purchase |

**Figure 8:** Most significant labels extracted through lda from files with their probabilistic contribution

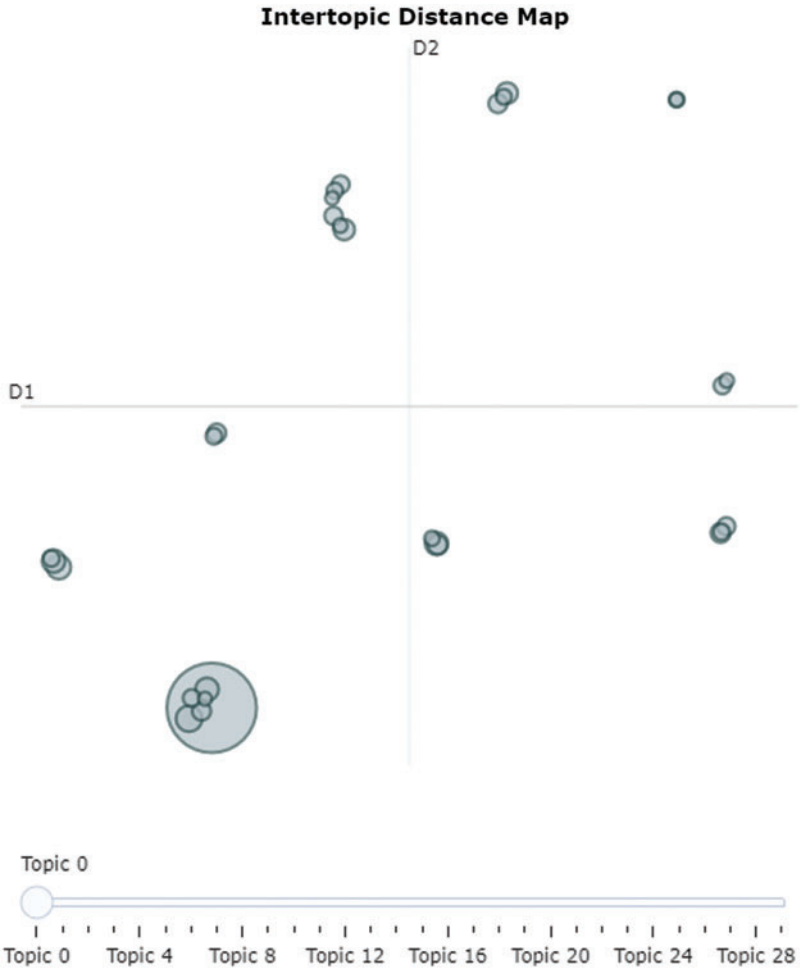
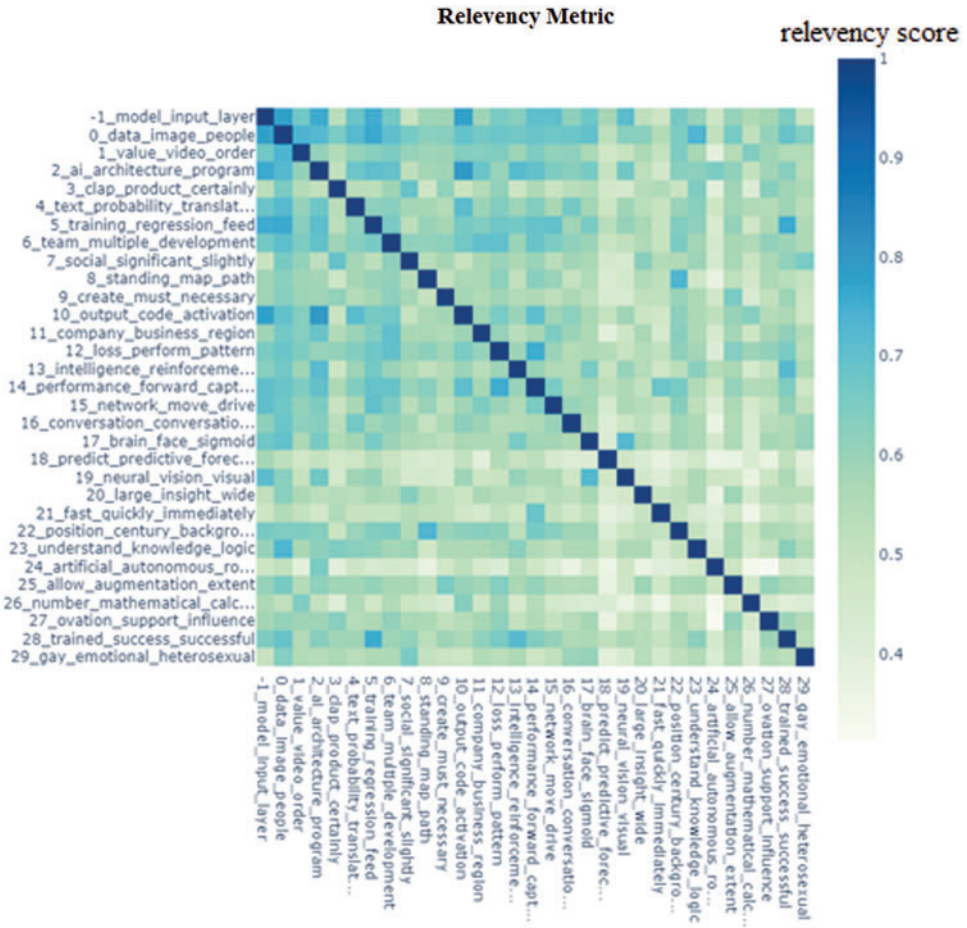



**Figure 9:** Intertopic distance map via multidimensional scaling of distinct labels

One prime objective of this research was to generate an automatic system independent of any human interaction. Therefore coherence and relevance metrics are used to identify common and significant terms and better understand the semantics of extracted words for the final label. One particular problem with statistical methods in textual analysis is that the selection of the words is generally based on the probabilities extracted based on word frequency. These approaches cannot analyze whether any word contributes to specific semantics or is just a most frequent but semantically less significant word. Carson et al. [32] established a relevance metric, which can rearrange the order of frequent words in a label by considering their relevancy to the file. This relevancy is measured through a weighting parameter $\theta$ that can range from 0 to 1 and ascribes the corpus frequencies. If the value of $\theta$ is close to 1, then the order of top words will be considered equal to the order of standard conditional probabilities, while $\theta$ closer to 0 reorder the most specific word to the top of the list. All the label words extracted from different topic modeling approaches were used to calculate the relevance, as mentioned in Fig. 10a.

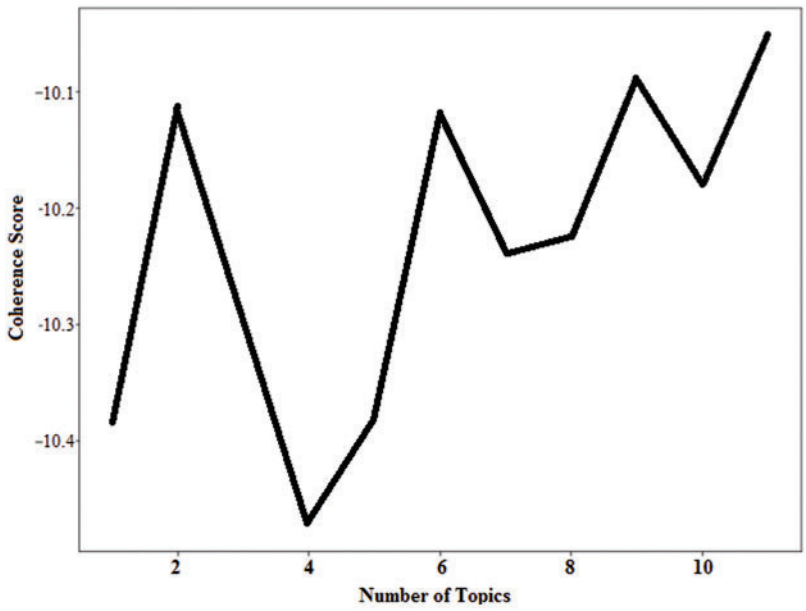
On the other hand, the coherence metric was developed by David et al. [33] and is being used for model selection and resolutions. However, it can help in guiding intuition when applied to single labels while identifying accurate labels in which a coherent concept might not be observed at first glance. Fig. 10b shows the coherence among the most relevant labels extracted through LDA, NMF, and LSA techniques. Finally, the relevant and most coherent topic words were given to BERT for file label generation. BERT was implemented with a fixed set of 3 words per label, as shown in Tab. 4.

It is clear from these results that the final label is much more meaningful than the label words extracted by LDA, NMF, and LSA individually. Moreover, these labels also give the impression of traditional file names given by a human. For instance, file 9 was a book on the history of the USA and is labeled as "Political history USA." The label for file 4, an article on intelligent agent architectures, is "AI architecture design." On the contrary, file 5 was an introductory brusher of different company products, and file 10 was an article on AI's informed and uninformed search techniques. Both of these files did not get any meaningful results. One primary reason for these non-meaningful labels is that these files were not textually rich and had multiple redundant terms instead of a complete textual theme.




(a) Relevancy Metric Results of extracted Labels

**Figure 10:** (Continued)




(b) Coherence Score for Extracted labels of Different Files

**Figure 10:** (a) Relevancy metric results of extracted labels (b) Coherence score for extracted labels of different files

**Table 4:** Updated file labels extracted using BERT

| File No. | Count | Label |
|---|---|---|
| 1 | 94461 | using_data_image |
| 2 | 30370 | model_classification_network |
| 3 | 8035 | video_order_price |
| 4 | 6802 | ai_architecture _design |
| 5 | 6283 | clap_product_certainly |
| 6 | 6185 | text_speech_translation |
| 7 | 6048 | feed_grid _regression |
| 8 | 5291 | multiple_team_development |
| 9 | 5145 | political_history_usa |
| 10 | 4465 | lower_path_map_ |
| 11 | 4270 | computational_code_activation |
| 12 | 4072 | region_business _industry |
| 13 | 4038 | pattern_loss_reduction |
| 14 | 3958 | intelligent_reinforcement_reward |
| 15 | 3743 | global_pollution_reduction |
| 16 | 3630 | network_drive_engine |

## 5 Conclusion

The increasing size and capacity of secondary storage devices and fast operating systems assist in aggregating the files users can store in their systems. However, this increase in the number of digital files might confuse users and cause them to spend additional time and effort finding and managing required files efficiently. Conversely, the need to automatically analyze, manage and label digital files has become significantly relevant. The particular challenge with file labeling is that it contains relatively heterogeneous and noisy data that might infer an inaccurate label. The proposed methodology can reasonably overcome this problem and can be used to label any textual file, including academic papers, user files, PowerPoint presentations, and PDFs. Despite individual label modeling techniques, qualitative evaluation of the sentence level BERT embedding reveals that these embeddings effectively organize a diverse range of digital files.

**Acknowledgement:** Thanks to our families & colleagues who supported us morally.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] J. D. Dinneen and J. Charles-Antoine, "The ubiquitous digital file: A review of file management research," *Journal of the Association for Information Science and Technology*, vol. 71, no. 1, pp. 1–32, 2020.

[2] C. John, "Creative names for personal files in an interactive computing environment," *International Journal of Man Machine Studies*, vol. 16, no. 4, pp. 405–438, 1982.

[3] H. Ben, D. Andy, R. Palmer and M. Hamish, "Organizing and managing personal electronic files: A mechanical engineer's perspective," *ACM Transactions on Information Systems*, vol. 26, no. 4, pp. 1–40, 2008.

[4] J. Crowder, S. M. Jonathan and R. Michele, "File naming in digital media research: Examples from the humanities and social sciences," *Journal of Librarianship and Scholarly Communication*, vol. 3, no. 3, pp. 1–22, 2015.

[5] T. Harumasa, H. Osamu and H. Masahiro, "A file naming scheme using hierarchical-keywords," in *26th Annual Int. Computer Software and Applications*, Oxford, England, pp. 799–804, 2002.

[6] L. Alon and N. Rafi, "Gaps between actual and ideal personal information management behavior," *Computers in Human Behavior*, vol. 107, no. 1, pp. 1–10, 2020.

[7] M. B. David, Y. N. Andrew and I. J. Michael, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 993–1022, 2003.

[8] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1, pp. 177–196, 2001.

[9] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, no. 1, pp. 556–562, 2001.

[10] D. A. Ostrowski, "Using latent dirichlet allocation for topic modelling in twitter," in *Proc. of the IEEE 9th Int. Conf. on Semantic Computing*, Anaheim, CA, USA, pp. 493–497, 2015.

[11] R. Daniel, R. Evan, C. Jason, D. M. Christopher and A. M. Daniel, "Topic modeling for the social sciences," in *NIPS 2009 Workshop on Applications for Topic Models: Text and Beyond*, Canada: Whistler, pp. 1–4, 2009.

[12] A. Rubayyi and A. Khalid, "A survey of topic modeling in text mining," *International Journal of Advanced Computer Science Application*, vol. 6, no. 1, pp. 147–153, 2015.

[13] P. Pantel and R. Deepak, "Automatically labeling semantic classes," in *Proc. of the Human Language Technology Conf. of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York, USA, pp. 321–328, 2004.
[14] A. Alokaili, N. Aletras and M. Stevenson, "Automatic generation of topic labels," in *Proc. of the 43rd Int. ACM Conf. on Research and Development in Information Retrieval*, Xi'an, China, pp. 1965–1968, 2020.
[15] J. Chang, G. Sean, W. Chong, B. G. Jordan and B. David, "Reading tea leaves: How humans interpret topic models," in *Twenty-Third Annual Conf. on Neural Information Processing Systems*, Vancouver, Canada, pp. 1–9, 2009.
[16] M. Qiaozhu, S. Xuehua and Z. Chengxiang, "Automatic labeling of multinomial topic models," in *Proc. of Thirteenth ACM Int. Conf. on Knowledge Discovery and Data Mining*, San Jose, California, pp. 490–499, 2007.
[17] H. Ioana, H. Conor, K. Marcel and G. Derek, "Unsupervised graph-based topic labelling using dbpedia," in *Proc. of Int. Conf. on Web Search and Data Mining*, Rome, Italy, pp. 465–473, 2013.
[18] M. Davide, C. Silvia, C. Davide and S. Fabio, "Automatic labeling of topics," in *Ninth Int. Conf. on Intelligent Systems Design and Applications*, Pisa, Italy, pp. 1227–1232, 2009.
[19] H. L. Jey, N. David, K. Sarvnaz and B. Timothy, "Best topic word selection for topic labelling," in *Proc. of the 23rd Int. Conf. on Computational Linguistics*, Beijing, China, pp. 605–613, 2010.
[20] W. Xiaojun and W. Tianming, "Automatic labeling of topic models using text summaries," in *Proc. of Association for Computational Linguistics*, Berlin, Germany, pp. 2297–2305, 2016.
[21] E. Amparo, B. Cano, H. Yulan and X. Ruifeng, "Automatic labelling of topic models learned from twitter by summarisation," in *Proc. of Association for Computational Linguistics*, Baltimore, USA, pp. 618–624, 2014.
[22] J. Hamed, W. Yongli, Y. Chi, F. Xia, J. Xiahui *et al.,* "Latent dirichlet allocation and topic modeling: Models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 15169–15211, 2019.
[23] D. Angelov, "Top2vec: Distributed representations of topics," *Arxiv*, vol. 22, pp. 1–10, 2022.
[24] G. Maarten, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *Arxiv*, vol. 22, pp. 1–13, 2022.
[25] S. B. Forrest, L. Hebi, L. Ge, C. Cen, Y. Yinfei *et al.,* "End-to-end semantics-based summary quality assessment for single-document summarization," in *Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, Seattle, Washington, USA, pp. 1–9, 2022.
[26] S. J. Karen, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 339–348, 1972.
[27] M. Tomas, C. Kai, C. Greg and D. Jeffrey, "Efficient estimation of word representations in vector space," in *1st Int. Conf. on Learning Representations*, Scottsdale, Arizona, USA, pp. 1–13, 2013.
[28] Q. Le and M. Tomas, "Distributed representations of sentences and documents," in *Int. Conf. on Machine Learning*, Beijing, China, pp. 1188–1196, 2014.
[29] G. N. Gopal, C. K. Binsu and U. Mini, "Keyword template based semi-supervised topic modelling in tweets," in *Int. Conf. on Innovative Computing and Communications*, Singapore, pp. 659–666, 2021.
[30] S. T. Dumais, "Latent semantic analysis," *Annual Review of Information Science and Technology*, vol. 38, no. 1, pp. 189–230, 2004.
[31] P. C. Hansen, "Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 3, pp. 503–518, 1990.
[32] S. Carson and S. Kenneth, "Ldavis: A method for visualizing and interpreting topics," in *Proc. of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, Baltimore, Maryland, USA, pp. 63–70, 2014.
[33] M. David, M. W. Hanna, T. Edmund, L. Miriam and M. Andrew, "Optimizing semantic coherence in topic models," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK, pp. 262–272, 2011.