Dynamic Discussion Topics Illustrator: a collective knowledge system for modeling social media topics evolution

Iulia-Maria Rădulescu [*], Alexandru Boicea [**], Mariana Mocanu [***], Florin Rădulescu [****] and Daniel Călin Popeangă [*****]
Computer Science and Engineering Department, Faculty of Automatic Control and Computer Science, Politehnica University of Bucharest, Romania, Bucharest

Abstract. Social media's ever-growing popularity led to the emergence of the Social Semantic Web, as an assembly of collective knowledge systems. This class of frameworks, algorithms, and tools aims to retrieve, process, and represent knowledge from human contributions. In this paper, we introduce a collective knowledge system to model the transformation of online conversations over time, allowing stakeholders to easily observe trends and behavior patterns. Our framework relies on an original, graph-based algorithm, called Dynamic Discussion Topics Illustrator, that builds "semantic evolutionary maps" of user discussion topics, which we call Discussion Topic Flows. The Discussion Topic Flows result from matching comment clusters from sequential time windows, according to their semantic similarity. The proposed system integrates the following phases: dataset preparation, text clustering, topic extraction, and finally, the employment of the Dynamic Discussion Topics Illustrator algorithm. We exemplify our method in a popular use case: automated extraction of user feedback from online software forums. For this purpose, we collect a real-world dataset of submissions posted on the Fedora dedicated subreddit: r/Fedora, over the entire year 2021. We evaluate the correctness of the results from three distinct perspectives: i) the comment clusters quality, assessed using three popular internal measures, ii) the Discussion Topic Flows' structure, expressed by their length and events quantity, and iii) the Discussion Topic Flows' explainability, measured through their comprising topics coherence.

Keywords: Social Semantic Web, Collective Knowledge Systems, Discussion Topic Flows, Text Clustering, Topic modeling, Alluvial Diagrams

1. Introduction

The Semantic Web is an extended version of the World Wide Web, where data is organized, linked, and represented in a standardized manner [1]. Intuitively, "the Semantic Web means sharing data and facts rather than sharing the text of a page" [1]. The rapid development of social media opened a new research direction in the context of the Semantic Web: the Social Semantic Web. The Social Semantic Web focuses on extracting, aggregating, and visualizing knowledge from social and conceptual networks [2]; it also studies the communication behaviors and

[*]Corresponding author. E-mail: iulia.m.radulescu@upb.ro.
[**]Corresponding author. E-mail: alexandru.boicea@upb.ro.
[***]Corresponding author. E-mail: mariana.mocanu@upb.ro.
[****]Corresponding author. E-mail: florin.radulescu@upb.ro.
[*****]Corresponding author. E-mail: daniel.popeanga@upb.ro.

the content and structure of user discussions [2]. Handling the temporal dimension of online communities, such as modeling changes over time in user interests, is notably important [3].

In this paper, we introduce a novel collective knowledge system, that builds topic maps of social media conversations relying on an original, graph-based algorithm, called Discussion Topics Illustrator. We model the topic transformations with the help of Discussion Topic Flows, a special type of data structure, formed of comment clusters from sequential time intervals. We thereby achieve the following **objectives**:

- design and implement an integrated system that models the temporal evolution of social media discussions;
- create an original algorithm for topic map generation;
- accelerate and simplify social media trends analysis process.

To demonstrate our method, we employ it for gathering user feedback from online forums: we analyze social media users' views on Fedora Linux, focusing on the release of the Fedora 35 distro (*2021-11-02*). We choose this use-case because almost all business organizations use online communities to improve their relationship with consumers [4]; at the same time, online user feedback has become a valuable resource of insights because the majority of users comment about the applications they use on E-commerce websites, social networks, and product forums [5]. For this purpose, we retrieve and process all the comments posted on the r/Fedora subreddit in 2021. For this real-world dataset, we assess both the accuracy of the clustering phase and the quality of the final Discussion Topic Flows by the number of insights they carry and their explainability.

The rest of this paper is organized as follows: in the Literature Review section (Section 2), we briefly describe several research articles regarding topic evolution, where various methods are employed to build topic maps; in Section 3 we detail all the concepts and techniques that build up our framework; here we present the Discussion Topics Illustrator algorithm in detail; in the Experimental Results (Section 5), we define the quality functions we make use of to evaluate Discussion Topic Flows, apply them, and discuss the results; finally, in Section 6, we conclude.

2. Background and Related Work

Methods for automating the extraction or enhancement of the structure of various text information is a core research topic in the Semantic Web domain [6]. However, unlike news or research articles corpora, social media streams put forward several challenges: they are short, noisy, and dynamic by nature—introduce the temporal dimension in addition to static text analysis [3].

Many literature studies from diverse domains analyze the evolution of online text information over time to extract valuable information. For example, regarding the recent Covid-19 pandemic, [7] build the flow of covid-related research topics to determine how such global crises impact the academic community and whether the impact is positive or negative. For this purpose, the authors employed the Scientific Evolutionary Pathways (SEP) algorithm [8].

Another application in the public health field is proposed by [9]: monitoring the evolution of vaccine-related topics to understand the general interests and concerns on the subject. The author identified the most prevalent pro and anti-vaccine topics by employing the Latent Dirichlet Allocation method on several online posts published between 2007 and 2017. To observe the interest flow for each topic, Zhan Xu maps its popularity (defined as the sum of all the shares, reactions, and comments of its comprising posts at six-month time intervals) to the temporal dimension.

[10] focus on an entirely different domain, investigating the evolution of radar research topics by employing a Latent Dirichlet Allocation-based dynamic topic model. The authors uncover the following insight: radar research progressively expanded its scope; even though it initially focused on a few topics, it became extremely popular over time, being applied in geography, meteorology, geology, ecology, and agriculture.

Researchers have designed numerous methods to capture topic evolution over time. Some approaches rely on statistics, such as the ones proposed by [11] and [12]; Blei et al. introduce Dynamic Topic Models (DTS), a generative model that extends Latent Dirichlet Allocation to handle temporal information. The authors organize documents into ordered time intervals; the topics from a particular time interval evolve from the topics linked to

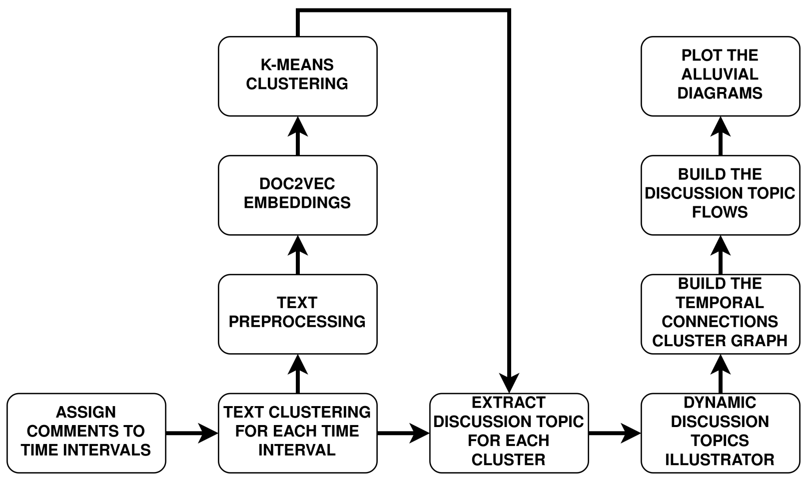


Fig. 1. The phases of the Discussion Topic Flows Framework: i) grouping comments into time intervals, according to their creation date, ii) clustering the comments of each time interval, iii) extracting the discussion topic of each cluster, and iv) feeding this information to the Dynamic Discussion Topics Illustrator algorithm.

the previous one [13]. Wang et al. develop another LDA-based generative model, called Topic Over Time (TOT); the model does not depend on time intervals, but rather uses topics to generate both words and timestamps for each document. Recent, more complex models, such as the Evolutionary Context-aware Sequential model (ECSM) built by [14], combine statistics with machine learning to discover topic evolutionary patterns. Other approaches, such as the SEP (Scientific Evolutionary Pathways) system created by [8], identify the relationships between topics with the help of science maps. SEP relies on text clustering and lays out the final result using graphs.

3. The Dynamic Discussion Topics Illustrator Knowledge System

The proposed collective knowledge system for building evolutionary topic maps consists of four iterative phases (which we display graphically in Fig. 1):

1. **batch setting**: organizing the user comments into sorted time intervals of equal size, according to their creation timestamp;
2. **identifying comment clusters**: the comments from each batch are clustered according to their textual similarity; this phase also incorporates the text pre-processing, embeddings generation, and dimension reduction processes;
3. **topic extraction**: extract the discussion topic of each comments cluster;
4. **employing the Dynamic Discussion Topics Illustrator algorithm**: the DDTI algorithm iterates the following three steps:

   (a) the comment clusters are matched into a directed graph, called Temporal Connections Cluster Graph; the Temporal Connections Cluster Graph models the overall temporal relationships between comment clusters from different time intervals;
   (b) building the Discussion Topic Flows by parsing the Temporal Connections Cluster Graph in a LIFO (Last In First Out) manner;
   (c) representing the previously computed Discussion Topic Flows visually with the help of alluvial diagrams; each cluster is labeled with the words that describe its latent discussion topic.

3.1. Text Clustering and Topic Extraction

This phase organizes the comments within a time interval into clusters, and incorporates the following steps: i) "cleaning up" comments by removing frequent words, links, and emojis, and reducing each word to its morphological, dictionary root, using the WordNet Lemmatizer, ii) transforming comments into numeric representations with the help of Doc2Vec embeddings, and iii) performing the actual clustering, by employing the KMeans clustering algorithm.

3.1.1. Text Preprocessing. WordNet Lemmatizer

The words in a text corpus may have several morphological forms, that are equivalent to the clustering process. To remove noise and reduce dimensionality, the derived words and the inflections are usually reduced to their base form using either stemming or lemmatization. Stemming algorithms truncate words by removing their suffixes, generally relying on a set of rules, while the lemmatization process computes the dictionary root of each word. For this reason, lemmatizers are more reliable than stemmers; therefore, we choose the WordNet Lemmatizer as the comments preprocessing method.

3.1.2. Document Embeddings. The Paragraph Vector Framework

A modern and accurate method to represent text corpora numerically is through neural network embeddings: they account for the semantic similarities between words and have a lower dimensionality than the *TF-IDF* frequency vectors. Several models for generating neural network embeddings have been developed across time: Paragraph Vector—Doc2Vec ([15]), Universal Sentence Encoder—USE ([16]), Sentence-BERT—S-BERT ([17]). The Paragraph Vector model (or Doc2Vec, for short), designed by [15], transforms varying length documents into vectors of a customizable number of dimensions. It relies on a simple neural network with a single layer of hidden neurons, and learns paragraph embeddings relying on two architectures: Distributed Memory Model (*PV-DM*) and Distributed Bag of Words (*PV-DBOW*). We choose the Doc2Vec model for transferring user comments to vectors due to its simplicity, and also because it allows specifying the size of the resulting embeddings (USE and S-BERT produce fixed-size, high-dimensional embeddings). When training Doc2Vec on the experimental dataset's text corpus, we prefer the *PV-DM* architecture, since the authors state that it is more accurate than *PV-DBOW* ([15]). We also employ hierarchical softmax instead of negative sampling.

To further reduce the dimensionality of the Doc2Vec embeddings, we employ the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) method, created by [18]. UMAP relies on manifold theory and topological data analysis: the algorithm first represents high-dimensional data as a fuzzy simplicial complex (as a weighted graph, where the weights represent the likelihood that two nodes are interconnected), then converts it into a low-dimensional representation, as structurally similar as possible. UMAP preserves more of the input dataset's global structure as compared with the state-of-the-art technique (t-SNE [19]) ([18]), and is also faster.

3.1.3. Clustering Algorithms. KMeans Clustering

We rely on the KMeans clustering algorithm [20] to group the semantically similar comment embeddings. The algorithm minimizes the total intra-cluster variance by repeating the following two steps until convergence: i) selecting k centroids $(c_i, \ldots, c_k)$ from the input dataset, either randomly or using a specific method, and ii) assigning the rest of the objects to k clusters, relying on their similarity to the previously selected centroids.

Because we cluster text data, we normalize the input embeddings before performing the actual clustering. We also normalize the resulting centroids before saving them as cluster attributes.

3.1.4. Topic extraction. Latent Dirichlet Allocation

We rely on Latent Dirichlet Allocation [21] to extract the most relevant words of each previously computed cluster. **Because clusters are already cohesive from the semantic point of view, each of them is characterized by a single discussion topic.**

Latent Dirichlet Allocation, or simpler, LDA, is a generative probabilistic model widely used to compute a text corpus's hidden (latent) topics. The base idea is that each text corpus contains several latent topics, and each topic is comprised of several words. More formal, documents are random mixtures over hidden topics, and each topic is characterized by a distribution over words [21].

For each cluster, we create a text corpus by merging all the comments within it. Then, we use it as input data for the LDA algorithm to retrieve the top k words forming its single latent topic.

4. Discussion Topic Flows. The Dynamic Discussion Topics Illustrator Algorithm

The Discussion Topic Flows (DTFs, for short) model the evolution of social media conversations over time, and represent the key concept of the proposed framework. They are formed of comment clusters from consecutive time intervals, that are interconnected with respect to their semantic relationships. To compute the Discussion Topic Flows from the time-interval clusters, we design an original, graph-based algorithm, called Dynamic Discussion Topics Illustrator (DDTI, for short).

The DDTI algorithm consists of three iterative phases:

- **building the Temporal Connections Cluster Graph**; the Temporal Connections Cluster Graph models the **overall semantic evolution of the users' discussions**; it interconnects all previously computed time-interval clusters, according to their centroids' similarities;
- **building the Discussion Topic Flows using the Temporal Connections Cluster Graph**; this phase segments the Temporal Connections Cluster Graph into discussion topic flows, by recursively parsing its nodes;
- **graphically illustrate the computed DTFs** with the help of alluvial diagrams.

In the following paragraphs, we describe each phase in detail.

4.1. Building the Temporal Connections Cluster Graph

We first represent each processed time interval as a directed cluster graph $G = (V, E)$, where the nodes are the comment clusters and the edges represent the euclidean similarities between the nodes. Each node is augmented with its comment cluster's centroid. Two nodes $v_1, v_2 \in V$ are interconnected by an edge if their centroids' euclidean similarity is greater than a predefined threshold θ: $1 - |v_1(centroid) - v_2(centroid)| > \theta$. **Each edge represents a match between two similar clusters from the same time interval.**

We then merge the time interval graphs into the Temporal Connections Cluster Graph. Two time interval graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are merged as follows: all vertices $\{v_1^1, \ldots, v_n^1\} \in V_1$ are interconnected with all vertices $\{v_1^2, \ldots, v_n^2\} \in V_2$ if their centroids' Euclidean similarity is greater than a predefined threshold θ. Each two vertices from **sequential time-intervals** are interconnected by **directed edges** to express the time flow (time cannot flow backwards). The resulting graph is merged with the next time interval graph. Therefore, the Temporal Connections Cluster Graph $CG = (V_{cg}, E_{cg})$ is defined as: $V_{cg} = V1 \cup V2$, $E_{cg} = \{(x_{cg}, y_{cg}) | (x_{cg}, y_{cg}) \in V_{cg}^2, 1 - |x_{cg}(centroid) - y_{cg}(centroid)| > \theta, x_{cg} \neq y_{cg}\}$. **Each edge represents a match between two similar clusters from sequential time intervals.** The pseudocode that describes the Temporal Connections Cluster Graph building process is displayed in Alg. 1.

4.2. Building the Discussion Topic Flows using the Temporal Connections Cluster Graph

The Temporal Connections Cluster Graph models the temporal evolution of all the discussion topics. In this second phase of the DDTI algorithm, we extract the Discussion Topic Flows from the Temporal Connections Cluster Graph by parsing it as follows: we initiate a Dynamic Topic Flow in each not already visited source node. We recursively extract and examine its destination nodes in a LIFO (Last In First Out) manner: each destination node is checked for other sources (since a cluster belonging to a certain time interval can match with more clusters from several other time intervals), then it is added to the Dynamic Topic Flow along with all its sources. The pseudocode that describes the Discussion Topic Flows building process is displayed in Alg. 2.

4.3. Visualization. Alluvial Diagrams

To represent the online conversations topic map visually, we plot the Discussion Topic Flows with the help of alluvial diagrams. Alluvial diagrams are flow diagrams, formed of several blocks interconnected by stream fields, used frequently to model a network's changes over time [22]. We exemplify our method's graphical output with the help of Fig. 2. Each block of the displayed alluvial diagram represents a comments cluster at a time interval. For example, the block labeled 1_45 stands for the second cluster from the 45th time interval (numbering starts at 0). **In**

Algorithm 1 Building the Temporal Connections Cluster Graph

```
procedure BUILDTEMPORALCONNECTIONSCLUSTERGRAPH(commentClustersForAllIntervals, θ)
    timeIntervalGraphs = ∅;
    for commentClustersForInterval ∈ commentClustersForAllIntervals do   ▷ build the clusters graph for each time interval;
        timeIntervalGraph;
        for commentCluster ∈ commentClustersForInterval do
            timeIntervalGraph.addNode(commentCluster);
        end for
        for commentCluster1 ∈ commentClustersForInterval do
            for commentCluster2 ∈ commentClustersForInterval do
                if 1 − |commentCluster1 − commentCluster2| > θ then
                    timeIntervalGraph.addEdge(commentCluster1, commentCluster2);
                end if
            end for
        end for
        timeIntervalGraphs = timeIntervalGraphs ∪ timeIntervalGraph;
    end for
    temporalConnectionsClusterGraph = merge(timeIntervalGraphs[0], timeIntervalGraphs[1]);
    for timeIntervalId ∈ [2, timeIntervalGraphs.length] do
        temporalConnectionsClusterGraph = merge(temporalConnectionsClusterGraph, timeIntervalGraphs[timeIntervalId]);
    end for
    return temporalConnectionsClusterGraph.adjacencyList;
end procedure
```

the Experimental Results section, we replace the labels with the most relevant 4 words of each cluster's main discussion topic, to improve the readability. The alluvial diagrams **capture the topic temporal relationships**, such as fusion or evolution; for example, the clusters 0, 1, 2 and 3 from the 45^{th} time interval (labeled 0_45, 1_45, 2_45, and 3_45) **fuse** into the second cluster at the next time interval (labeled 1_48). At the same time, cluster 1_45 **evolves** into the second cluster at the 49^{th} time interval.

5. Experimental Results

In this section, we apply the proposed framework to a collection of Reddit comments regarding Fedora Linux, posted in 2021. We build the DTFs **for the whole year**, as well as for the interval: ***2021-11-01* to *2021-12-31* (2 months after the release of the new Fedora 35 distro)**—the longest DTFs corresponding to $\theta = 75\%$, $\theta = 80\%$, $\theta = 85\%$, $\theta = 90\%$ similarity thresholds being displayed in Fig. 3.

We evaluate the DTF method's results from three perspectives: i) **the comment clusters quality** (Subsection 5.4.1), ii) **the correctness of the cluster matching operation** (Subsection 5.4.2), and iii) **the coherence of the topics comprising the Discussion Topic Flows** (Subsection 5.4.3), with the help of the quality functions described in Subsection 5.3.

We present the insights retrieved by making use of the proposed system in Subsection 5.4.4.

5.1. Experimental Setup

We perform the following experiments on an Oracle Linux Server machine with 40 CPUs and 64 GB RAM. The code is entirely implemented in Python 3.8.12 and is available online on Github.[1] For the text preprocessing oper-

[1]https://github.com/IuliaRadulescu/DynamicTextAnalysis

Algorithm 2 Building the Discussion Topic Flows

```
procedure BUILDDYNAMICTOPICFLOWSFROMGRAPH(graphAdjacencyList, startInterval, endInterval)
    for node ∈ graphAdjacencyList do                ▷ filter only desired time interval from graph;
        if timeInterval(node) < startInterval or timeInterval(node) > endInterval then
            graphAdjacencyList = graphAdjacencyList \ node;
        end if
    end for
    reversedGraphAdjacencyList = reverse(graphAdjacencyList);  ▷ reverse the clusters graph adjacency list to ease retrieval of source nodes (flip source nodes with destination nodes);
    parsedNodes = ∅;
    dynamicTopicFlows = ∅;
    for node ∈ graphAdjacencyList do
        dynamicTopicFlow = ∅;
        if node in parsedNodes then continue;
        end if
        dynamicTopicFlow = checkNodeForMatches(graphAdjacencyList[node], reversedGraphAdjacencyList, parsedNodes, dynamicTopicFlow);
        nodesStack = graphAdjacencyList[node];
        while nodesStack ≠ ∅ do
            stackNode = nodesStack.pop();
            dynamicTopicFlow = checkNodeForMatches(graphAdjacencyList[stackNode], reversedGraphAdjacencyList, parsedNodes, dynamicTopicFlow);
        end while
        dynamicTopicFlows = dynamicTopicFlows ∪ dynamicTopicFlow
    end for
    return dynamicTopicFlows
end procedure
procedure CHECKNODEFORMATCHES(destinationNodes, reversedGraphAdjacencyList, parsedNodes, dynamicTopicFlow)
    for destinationNode ∈ destinationNodes do
        matchingSources = reversedGraphAdjacencyList[destinationNode] \ parsedNodes;
        for matchingSource ∈ matchingSources do
            dynamicTopicFlow = dynamicTopicFlow ∪ matchingSource;
        end for

    end for
end procedure
```

ations (stop words removal, tokenization and lemmatization), we apply the methods provided by the *nltk* [2] library. We use the *scikit-learn* [3] implementation of the KMeans Clustering and Latent Dirichlet Allocation algorithms. We calculate the correct number of clusters (ranging between 2 and 20) by applying the Silhouette method. As for representing user comments numerically, we rely on the Doc2Vec model provided by *gensim* [4], setting the hidden layer size to 16 neurons, and the window size to 3. We reduce the number of dimensions to 3 employing *UMAP* [5].

[2] https://www.nltk.org/
[3] https://scikit-learn.org/stable/
[4] https://radimrehurek.com/gensim/index.html
[5] https://umap-learn.readthedocs.io/en/latest/

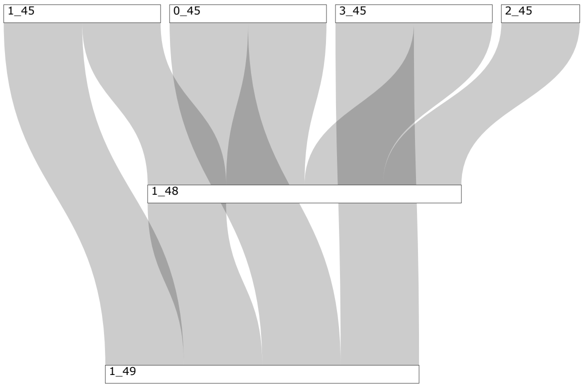


Fig. 2. Alluvial diagram example

We use the coherence measures provided by *Palmetto* [6] (using our own text-corpus-generated index) and *gensim* to evaluate the DTF topics' quality. To plot the results as alluvial diagrams, we employ the *plotly* [7] library.

5.2. Dataset

We apply and evaluate our framework on a real-world dataset consisting of 22 204 comments, from 4 554 unique users, for 6 689 submissions posted on the Fedora dedicated subreddit: r/Fedora, extracted using two free APIs: i) the Pushift API [8] and ii) PRAW [9]. Each element of the dataset contains: i) the text information of the comment, ii) its author's user id, and iii) its creation timestamp. We retrieved all the comments posted in 2021. We aim to model the general feedback as well as fresh user reactions to Fedora 35's release, which took place on the 2nd of November 2021, employing the DTF method. When grouping the comments into batches according to their creation timestamp, we use a 1-week interval to allow network snapshots to share more nodes. However, none of the network snapshots shares any nodes. The selected dataset is suitable for our experiments because, even for a large time window, the community network structure changes considerably.

5.3. Evaluation Measures

In the following subsections, we briefly describe the quality functions that we apply to validate the proposed framework.

5.3.1. Text Clusters Evaluation

To evaluate the text clusters quality we use several well-known internal evaluation measures: the Calinski-Harabasz Index (CH Index) [23], the Davies-Bouldin Index (DB Index) [24], and the Silhouette Coefficient (Silh Coefficient) [25]. We are using only internal evaluation measures because we run our experiments on a real-world dataset without pre-defined labels.

The Calinski-Harabasz Index (CH Index) [23] measures the average dispersion degree for all the clusters. A large value for the CH Index indicates well-separated clusters because it is proportional to the inter-cluster distance [26].

[6] https://github.com/dice-group/Palmetto
[7] https://plotly.com/
[8] https://github.com/pushshift/api
[9] https://praw.readthedocs.io/en/stable/

The Silhouette Coefficient(Silh Coefficient) [25] measures the degree of membership of an object to its cluster [26] and is computed for each object in the dataset. The Silhouette Coefficient values fall in the interval $[-1, 1]$; if the Silhouette Coefficient of an object is close to 1, that object has high cluster membership. We average the Silhouette Coefficient values for each object in the input dataset.

The Davies-Bouldin Index (DB Index) [24] measures a cluster's compactness and separation [27] by calculating the ratio between the intra-cluster and inter-cluster distances. The smaller the value of the DB Index, the better separated the clusters.

5.3.2. Discussion Topic Flows Evaluation—Clusters Matching

We determine the DDTI's algorithm correctness by examining the way the number of matched clusters and the size of the topic evolution flows vary with the cluster similarity threshold θ. For this purpose, we view the sets of matched clusters as dynamic communities and characterize them by defining two evaluation measures inspired from dynamic community detection [28]: i) the Width (average and maximum) of all the topic evolution flows and ii) the Length (average and maximum) of all the topic evolution flows.

The Width of a single DTF is expressed as the number cluster matches within that DTF. The average, respectively maximum Width of several DTFs are formally defined in Def 1.

Definition 1 (Definition of Width)**.** *The average, respectively maximum Width of a set* $DTF = (DTF_1, \ldots, DTF_n)$ *of n Discussion Topic Flows are formally expressed as:*
i) $W_{avg} = \overline{n_{matched}(C_{DTF_k})}, k \in \{1, \ldots, n\}$*, and*
ii) $W_{max} = \max(n_{matched}(C_{DTF_1}), n_{matched}(C_{DTF_2}), \ldots, n_{matched}(C_{DTF_n}))$*, where* $n_{matched}(C_{DTF_k})$ *is the number of matched clusters within* DTF_k*.*

The Length of a single DTF is expressed as the number of distinct time intervals it spans. The average, respectively maximum Length of several DTFs are formally defined in Def 2.

Definition 2 (Definition of Length)**.** *The average, respectively maximum Length of a set* $DTF = (DTF_1, \ldots, DTF_n)$ *of n Discussion Topic Flows are formally expressed as:*
i) $L_{avg} = \overline{L_D}$*, and*
ii) $L_{max} = \max(L_D)$*, where* L_D *is the list of dynamic topic flow lengths* $(L_{DTF_1}, L_{DTF_2}, \ldots, L_{DTF_n})$*. The length of a dynamic topic flow* L_{DTF_k} *is given by the number* $n(T)$ *of distinct intervals it spans on:* $T = \{t_x^{DTF_k} | x \in \mathbb{N}\}$*.*

5.3.3. Discussion Topic Flows Evaluation—Topic Coherence

Topic coherence measures how easy to interpret the topics generated by a topic model are to humans ([29]). Since the primary purpose of the Discussion Topic Flows is to be analyzed by software stakeholders to extract insights regarding users' feedback, **topic coherence is suitable for assessing the quality of their comprising topics**. [30] build a flexible framework that unifies word-based coherence measures through the combination of four modules: i) the Segmentation module, which creates several topic-word subsets, ii) the Probability Calculation module, which defines the method for computing the text corpus probabilities, iii) the Confirmation Measure module, that determines the association between the topic-word subsets using the probabilities from the previous step, and iv) the Aggregation module, that accumulates the coherence measures of all topic-word subsets into the final coherence score. The Confirmation Measure module allows the use of both direct and indirect confirmation measures. The direct confirmation measures depend only on the topic-word subsets and the probabilities derived from the text-corpus, while indirect confirmation measures rely on vectors of direct measures. As a result, indirect confirmation measures can capture more complex word relationships than direct measures ([30]). Thus, **we prefer direct confirmation measures to validate our results**. Röder et al. evaluate several topic coherence measures, that they first implement with the help of their unifying framework. The best performing indirect measure is the C_V measure. C_V pairs each word in a topic with that topic's entire set of words in the Segmentation phase and uses a boolean sliding window probability to compute the text corpus probabilities; as for comparing the direct coherence vectors, it employs the cosine similarity. Its values range from -1 to 1; **a value close to 1 denotes high coherence**. We assess the quality of the topics within the computed DTFs with the help of the C_V **coherence measure**. **Since our text corpus is formed of short comments, we set the sliding window size to 1.** We mention that even though C_V encounters problems for

Table 1
Clustering evaluation results for the Fedora 35 comments dataset versus several edge cases: random, unbalanced, and isolated

	CH	CH Random	CH Unbalanced	DB	DB Random	DB Unbalanced	Silh	Silh Random	Silh Unbalanced
Minimum	62.813	0.699	0.190	0.677	4.178	0.654	0.584	-0.66	-0.528
Average	230.858	0.986	0.99	1.206	6.37	1.03	0.673	-0.623	-0.156
Maximum	599.069	1.153	2.085	1.546	8.617	2.153	0.815	-0.567	0.326

randomly generated word sets (as Röder et al. state [10]), **our dataset is appropriate for this coherence measure**, because it contains cleaned and lemmatized social media comments. Moreover, we validate the C_V coherence's values by comparing them to the ones obtained by the *UMASS* coherence. *UMASS* is an asymmetric measure, designed by [31], that estimates word probabilities based on the document frequencies of the text corpus [30].

5.4. Results Discussion

5.4.1. Comment Clusters Quality Assessment

First, we analyze the clustering phase's results, using the quality functions defined in Subsection 5.3. To better understand their values, we also compute the DB Index, CH Index, and Silh Coefficient for two edge cases: i) **the random case**, where each comment is randomly assigned to a cluster and ii) **the unbalanced case**, were a single comment is assigned to one cluster, while the rest of the comments form a large cluster. We display the computed values in Table 1.

We observe that the average values for the CH Index significantly differ between the real assignment and the edge scenarios: 230.858 versus 0.986 for the random case and 0.99 for the unbalanced case. The smallest CH Index value for the real assignment is 62.813, while the largest is approximately 600. Since the CH Index must be large, the values indicate well-separated, dense clusters.

The mean Silh Coefficient corresponding to the real assignment is 0.673, while the minimum and maximum values are 0.584, respectively 0.815. All the values are above 0.5, thus, overall, the clusters are relatively dense and well-separated. The negative values for the edge cases indicate that the comments were assigned to the incorrect clusters.

The mean DB Index of the real clusters is 1.206, smaller than the one corresponding to the random case: 6.37. The unbalanced case yields a smaller value: 1.03; however, this is because the ratio between intra-cluster and inter-cluster distances strongly depends on which comment is left out (it can significantly differ from the rest of the dataset).

5.4.2. Discussion Topic Flows Quality Assessment—Clusters Matching

Next, we validate the correctness of the Discussion Topic Flows considering their activity (the number of events they hold). For this purpose, we rely on their Width and Length, defined in Subsection 5.3.2. We display the evaluation results for the full dataset, as well as for the release subset, with several cluster similarity thresholds applied ($\theta = [75\%, 80\%, 85\%, 90\%]$) in Tables 2 and 3.

We start with the DTFs for the entire year 2021. We observe that the maximum Width of the Discussion Topic Flows decreases as the cluster similarity threshold increases. This behavior is natural and validates the correctness of our method since the Width measures the number of matched clusters (and for higher values of θ only a few clusters match—those that are extremely similar). The decrease is best visible for the maximum Width values: 2327 for $\theta = 75\%$, 1438 for $\theta = 80\%$, 693 for $\theta = 85\%$, 140 for $\theta = 90\%$. Similarly, the DTFs' maximum Length decreases with the centroid similarity threshold's growth. The maximum Length ranges from 49 for $\theta = 75\%$ to 21 for $\theta = 90\%$. This is because, when fewer clusters match, the Discussion Topic Flows are shorter. We observe that there is a significant difference between the maximum, versus mean Width and Length. For example, for $\theta = 75\%$, the maximum number of matches is 2327, while the mean number of matches is 246. This is because all DTFs contain many one-to-one matches, that interconnect only two topics. These connections are specific to short-lived

[10]https://github.com/dice-group/Palmetto/issues/13#issuecomment-371553052

Table 2
Cluster matching results for the Fedora 35 comments dataset, in 2021, using different cluster similarities

Centroid similarity	Width		Length	
	Max	Mean	Max	Mean
75% similarity	2327	246	49	15.25
80% similarity	1438	145.41	49	13.35
85% similarity	693	55.76	47	8.76
90% similarity	140	17.39	21	6.42

Table 3
Cluster matching results for the Fedora 35 comments dataset, immediately after the release (between *2021-11-01* and *2021-12-31*), using different cluster similarities

Centroid similarity	Width		Length	
	Max	Mean	Max	Mean
75% similarity	91	18	9	4
80% similarity	54	12.4	9	4.1
85% similarity	21	8.42	5	3.85
90% similarity	8	3.4	3	2.5

discussions, and the mean Width and Length are strongly influenced by topics that matched one or two fronts. **For small θ values (75 and below), many noisy matches are made. Otherwise, if θ is too large (90 and above), it filters most of the matches, thus reducing the insights provided by the DTF**.

The Width and Length for the 2-month interval following Fedora 35's release keep to the same pattern. One notable observation is that the difference between the Maximum and Mean values is smaller due to fewer one-to-one matches in a shorter time (2 months instead of 365 days).

5.4.3. Discussion Topic Flows Quality Assessment—Topics Quality

We verify the explainability of the Discussion Topic Flows using the C_V coherence measure. We also compute the *UMASS* coherence to ensure the validity of the results. We display the C_V and *UMASS* values for both entire year and release subset in Table 4. We analyze the longest-lived DTFs for each similarity threshold since they are the most complex and carry the most information.

We first observe that, for both time intervals, we obtain high C_V scores (all above 0.89). The *UMASS* measure is positively correlated with the C_V coherence, especially for the release subset, where it decreases from -2.33 ($\theta = 75\%$) to -1.83 ($\theta = 95\%$) as C_V increases (UMASS values close to 0 indicate high coherence [32], thus the correlation is positive). The results for the release subset also show that the DTFs corresponding to lower similarity thresholds are less intuitive (the lowest C_V value of the release subset is 0.896 for $\theta = 75\%$, while the highest is 0.907 for $\theta = 90\%$). This is because small θ values allow matches between less similar centroids, thus, the generated flows are formed of semantically unrelated discussion topics. However, even though the topic flows corresponding to large similarity thresholds ($\theta = 90\%$, $\theta = 95\%$) are characterized by high coherence values, they are short-lived and contain few matches (as we show in the previous section). Hence, **the best DTFs must contain a balanced number of cluster matches: too many matches cause inaccurate topic flows, while few matches reduce the provided insights.** The number of clusters matched is controlled by the similarity threshold.

5.4.4. Insights

Each of the resulting Discussion Topic Flows represents the "evolution map" of user conversations, that offers interesting insights, such as the impact of new features and reactions to bugs/ hotfixes.

The alluvial diagrams that model the overall activity are very complex, due to the large number of simple matches that interconnect one or two comment clusters. The longest-lived DTFs for the the interval *2021-11-01* to *2021-12-31* are more intuitive and therefore can be analyzed directly. We lay them out in Fig. 3. In the following paragraphs,

Table 4

Topic coherence for the longest-lived DTFs, for both entire year 2021 and immediately after the release (between *2021-11-01* and *2021-12-31*), using different cluster similarities

	Overall topic coherence			
	75%	80%	85%	90%
C_V Full 2021	0.896	0.896	0.898	0.898
UMASS Full 2021	-2.26	-2.26	-2.23	-2.25
C_V November - December 2021	0.896	0.897	0.899	0.907
UMASS November - December 2021	-2.33	-2.27	-2.00	-1.83

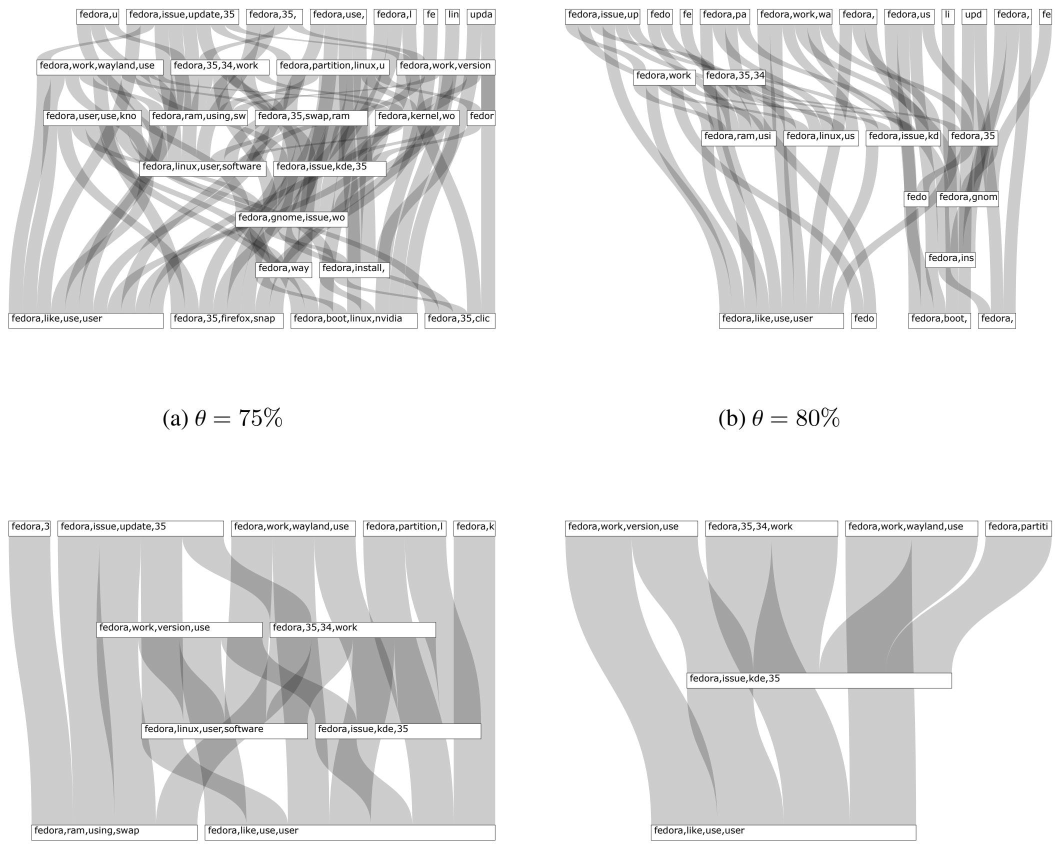
we examine the DTFs corresponding to $\theta = 85$ and $\theta = 80$ to show how user feedback regarding the release of the Fedora 35 changed over 2 months. According to the previous sections (Section 5.4.2 and Section 5.4.3), they are the easiest to interpret (have high coherence, above 0.8) and also contain potentially relevant insights (are long-lived and contain enough matches). **We also display the corresponding Discussion Topic Flows in text format in Table 5** to improve readability.

The most prevalent topic words within the DTF, aside Fedora, are: **34**, **35**, and **update**. This is natural, since the analyzed time interval follows Fedora 35's release, thus, many users posted about updating their distro from version 34 to version 35. We observe that the word **issue** appears often near **update**, implying that the users ran into problems during the update. We then notice that the words **issue** and **kde** are correlated. As Fedora 35 includes a new KDE version (KDE Plasma 5.22 [11]), the DTF indicates a bug related to it. Drilling down and reading the complete comments, **we discover that the assumption is true**; for example, the following comment, posted 8 months ago, publicly available online, states: *KDE on Fedora is quite buggy. I'm on Fedora 34 as of 4 days ago and I've met some weirdness. For example, system settings crashed twice while I was creating a custom keyboard shortcut. I've set some custom shortcuts but one of them is not working....* [12]

From the discussion flow perspective, 8 out of 9 discussions referring update problems (*fedora issue update 35*) evolve into conversations about fixes attempts and suggestions (*fedora work version use* and *fedora 35 34 work*, that are comprised of comments such as: *it will work fine if you can get the rhel rpm and extract the contents, i also got the flatpak to work as well, how to get this to work*). Then, each flow path ends with a specific topic (*fedora like use user*, *fedora ram using swap*, *fedora issue kde 35*). We also notice that the discussions that are not directly concerning issues (*fedora work wayland use*, *fedora partition linux use*) merge into the conversation about the kde bugs (*fedora issue kde 35*). Therefore, the issues with Fedora 35 impacted the users immediately after the release, and they turned to online communities to find fixes/ workarounds.

The DTFs corresponding to lower similarity thresholds include the discussion topics of the ones for higher similarity thresholds; thus, the DTF for $\theta = 80\%$ repeats the previously mentioned patterns, while also adding new information. The new conversations revolve around Gnome, a graphical user interface for Linux systems. Fedora 35 ships with a new Gnome version (Gnome 41) [13]. Gnome 41 introduces several improvements [14], such as GNOME Software Center user interface enhancements. The DTF indicates that these changes were noticed and generated discussions: **gnome** is a prevalent topic word, while the word **software** is also mentioned. The words **kde**, **click**, and **middle** relate to the previously mentioned user interface issues. The discussion transformation paths also contain references to Wayland and Nvidia. Wayland is a display server protocol offered by default by many Linux distros, including Fedora [15]. The compatibility between Wayland and Nvidia graphics cards is a known issue [16]. Knowing that this configuration is error-prone, one could drill down and read the actual comments to gather more information.

[11] https://news.itsfoss.com/kde-plasma-5-22-release/
[12] https://www.reddit.com/r/Fedora/comments/q94q95/fedora_kde_bugs/
[13] https://news.itsfoss.com/fedora-35-release/
[14] https://news.itsfoss.com/gnome-41-release/
[15] https://www.makeuseof.com/tag/using-linux-with-wayland/
[16] https://www.makeuseof.com/tag/using-linux-with-wayland/



(a) $\theta = 75\%$ (b) $\theta = 80\%$

(c) $\theta = 85\%$ (d) $\theta = 90\%$

Fig. 3. The longest Discussion Topic Flows representing the topic evolution for all similarity thresholds, after Fedora 35's release (*2021-11-01* to *2021-12-31*)

6. Conclusions

In this research paper, we described and evaluated a novel collective knowledge system for extracting insights from social media conversations, based on an original algorithm called Dynamic Discussion Topics Illustrator. We combine clustering, topic modeling, and graph processing to build "semantic maps", called Discussion Topic Flows, from online comment streams.

We applied the proposed framework to respond to an up-to-date challenge: gathering software users' feedback from online communities, specifically, Reddit users' views on Fedora Linux, over the entire year 2021 and immediately after Fedora 35 distro's release. For example, we identified the main reasons for dissatisfaction reported by the users: graphical display bugs related to KDE, and most important, Wayland compatibility with Nvidia. We modeled the users' feedback topic map graphically, in an intuitive and easy-to-follow manner.

In the future, we intend to enhance the DDTI algorithm as follows: i) improve the topic coherence of the Discussion Topic Flows: they currently include noisy short-lived discussions, that last only a single time interval and reduce the explainability, ii) create simpler Discussion Topic Flows: the DTFs corresponding to the full dataset are complex, thus, hard to visualize as static images, iii) represent each topic by sentences instead the top *k* words, and iv) account for users' emotions; it would be useful to differentiate between positive and negative feedback.

Table 5

The topic evolution paths for $\theta = 85\%$ and $\theta = 80\%$ similarity thresholds. As the pathways of lower similarity thresholds include the pathways of higher ones, we only display the differences for $\theta = 85\%$.

85%	fedora 35 update 34 → fedora ram using swap
	fedora issue update 35 → fedora 35 34 work → fedora issue kde 35 → fedora like use user
	fedora issue update 35 → fedora 35 34 work → fedora like use user
	fedora issue update 35 → fedora 35 34 work → fedora linux user software → fedora like use user
	fedora issue update 35 → fedora 35 34 work → fedora ram using swap
	fedora issue update 35 → fedora linux user software → fedora ram using swap
	fedora issue update 35 → fedora work version use → fedora issue kde 35 → fedora like use user
	fedora issue update 35 → fedora work version use → fedora like use user
	fedora issue update 35 → fedora work version use → fedora linux user software → fedora like use user
	fedora issue update 35 → fedora work version use → fedora ram using swap
	fedora kernel work wifi → fedora like use user
	fedora partition linux use → fedora issue kde 35 → fedora like use user
	fedora partition linux use → fedora like use user
	fedora work wayland use → fedora issue kde 35 → fedora like use user
	fedora work wayland use → fedora like use user
	fedora work wayland use → fedora linux user software → fedora like use user
80%	fedora 35 update 34 → fedora ram using swap → fedora like use user
	fedora issue update 35 → fedora 35 34 work → fedora 35 swap ram → fedora gnome issue work → fedora install boot work → fedora 35 click middle
	fedora issue update 35 → fedora 35 34 work → fedora 35 swap ram → fedora gnome issue work → fedora install boot work → fedora boot linux nvidia
	fedora issue update 35 → fedora 35 34 work → fedora 35 swap ram → fedora like use user
	fedora issue update 35 → fedora 35 34 work → fedora 35 swap ram → fedora wayland 35 gnome → fedora boot linux nvidia
	fedora issue update 35 → fedora 35 34 work → fedora ram using swap → fedora like use user
	fedora issue update 35 → fedora issue kde 35 → fedora linux user software → fedora ram using swap → fedora like use user
	fedora issue update 35 → fedora work version use → fedora gnome like install
	fedora issue update 35 → fedora work version use → fedora ram using swap → fedora like use user
	fedora kernel work wifi → fedora gnome issue work → fedora install boot work → fedora 35 click middle
	fedora kernel work wifi → fedora gnome issue work → fedora install boot work → fedora boot linux nvidia
	fedora kernel work wifi → fedora issue kde 35 → fedora like use user
	fedora linux kernel amazon → fedora gnome issue work → fedora install boot work → fedora 35 click middle
	fedora linux kernel amazon → fedora gnome issue work → fedora install boot work → fedora boot linux nvidia
	fedora partition linux use → fedora gnome like install
	fedora partition linux use → fedora linux user software → fedora like use user
	fedora user use know → fedora gnome issue work → fedora install boot work → fedora 35 click middle
	fedora user use know → fedora gnome issue work → fedora install boot work → fedora boot linux nvidia
	fedora user use know → fedora install boot work → fedora 35 click middle
	fedora use work like → fedora 35 swap ram → fedora gnome issue work → fedora install boot work → fedora 35 click middle
	fedora use work like → fedora 35 swap ram → fedora gnome issue work → fedora install boot work → fedora boot linux nvidia
	fedora use work like → fedora 35 swap ram → fedora like use user
	fedora use work like → fedora 35 swap ram → fedora wayland 35 gnome → fedora boot linux nvidia
	fedora use work like → fedora wayland 35 gnome → fedora install boot work → fedora boot linux nvidia
	fedora use work window → fedora linux user software → fedora like use user
	fedora use work window → fedora ram using swap → fedora like use user
	fedora work wayland use → fedora 35 swap ram → fedora gnome issue work → fedora install boot work → fedora 35 click middle
	fedora work wayland use → fedora 35 swap ram → fedora gnome issue work → fedora install boot work → fedora boot linux nvidia
	fedora work wayland use → fedora 35 swap ram → fedora like use user
	fedora work wayland use → fedora 35 swap ram → fedora wayland 35 gnome → fedora boot linux nvidia
	fedora work wayland use → fedora ram using swap → fedora like use user
	linux fedora gnome window → fedora boot linux nvidia
	update fedora system use → fedora 35 click middle
	update fedora system use → fedora boot linux nvidia

References

[1] A. Patel and S. Jain, Present and future of semantic web technologies: a research statement, *International Journal of Computers and Applications* **43**(5) (2021), 413–422.

[2] F. Gandon, A survey of the first 20 years of research on semantic Web and linked data, *Revue des Sciences et Technologies de l'Information-Série ISI: Ingénierie des Systèmes d'information* (2018).

[3] K. Bontcheva and D. Rout, Making sense of social media streams through semantics: a survey, *Semantic Web* **5**(5) (2014), 373–403.

[4] V. Sundararaj and M. Rejeesh, A detailed behavioral analysis on consumer and customer changing behavior with respect to social networking sites, *Journal of Retailing and Consumer Services* **58** (2021), 102190.
[5] J. Tizard, T. Rietz, X. Liu and K. Blincoe, Voice of the users: an extended study of software feedback engagement, *Requirements Engineering* (2021), 1–23.
[6] J.L. Martinez-Rodriguez, A. Hogan and I. Lopez-Arevalo, Information extraction meets the semantic web: a survey, *Semantic Web* **11**(2) (2020), 255–335.
[7] Y. Zhang, X. Cai, C.V. Fry, M. Wu and C.S. Wagner, Topic evolution, disruption and resilience in early COVID-19 research, *Scientometrics* **126**(5) (2021), 4225–4253.
[8] Y. Zhang, G. Zhang, D. Zhu and J. Lu, Scientific evolutionary pathways: Identifying and visualizing relationships for scientific topics, *Journal of the Association for Information Science and Technology* **68**(8) (2017), 1925–1939.
[9] Z. Xu, Personal stories matter: topic evolution and popularity among pro-and anti-vaccine online articles, *Journal of computational social science* **2**(2) (2019), 207–220.
[10] X. Huang and H. Fang, Topic evolution analysis of radar research using a dynamic topic model based on latent Dirichlet allocation, in: *Journal of Physics: Conference Series*, Vol. 2010, IOP Publishing, 2021, p. 012105.
[11] D.M. Blei and J.D. Lafferty, Topic models, in: *Text mining*, Chapman and Hall/CRC, 2009, pp. 101–124.
[12] X. Wang and A. McCallum, Topics over time: a non-markov continuous-time model of topical trends, in: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 424–433.
[13] L.E. George and L. Birla, A study of topic modeling methods, in: *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2018, pp. 109–113.
[14] Z. Lu, H. Tan and W. Li, An evolutionary context-aware sequential model for topic evolution of text stream, *Information Sciences* **473** (2019), 166–177.
[15] Q. Le and T. Mikolov, Distributed representations of sentences and documents, in: *ICML*, 2014, pp. 1188–1196.
[16] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R.S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar et al., Universal sentence encoder for English, in: *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, 2018, pp. 169–174.
[17] N. Reimers and I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.
[18] L. McInnes, J. Healy and J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, *arXiv preprint arXiv:1802.03426* (2018).
[19] L. Van der Maaten and G. Hinton, Visualizing data using t-SNE., *Journal of machine learning research* **9**(11) (2008).
[20] J. MacQueen, Classification and analysis of multivariate observations, in: *5th Berkeley Symp. Math. Statist. Probability*, 1967, pp. 281–297.
[21] D.M. Blei, A.Y. Ng and M.I. Jordan, Latent dirichlet allocation, *the Journal of machine Learning research* **3** (2003), 993–1022.
[22] A.W.K. Yeung, Data visualization by alluvial diagrams for bibliometric reports, systematic reviews and meta-analyses, *Current Science* (2018).
[23] T. Caliński and J. Harabasz, A dendrite method for cluster analysis, *Communications in Statistics-theory and Methods* **3**(1) (1974), 1–27.
[24] D.L. Davies and D.W. Bouldin, A cluster separation measure, *IEEE transactions on pattern analysis and machine intelligence* (1979), 224–227.
[25] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics* **20** (1987), 53–65.
[26] L. Cui, X. Song and G. Zhong, Comparative Analysis of Three Methods for HYSPLIT Atmospheric Trajectories Clustering, *Atmosphere* **12**(6) (2021), 698.
[27] Y. Liu, Z. Li, H. Xiong, X. Gao and J. Wu, Understanding of internal clustering validation measures, in: *2010 IEEE international conference on data mining*, IEEE, 2010, pp. 911–916.
[28] I.X. Leung, P. Hui, P. Lio and J. Crowcroft, Towards real-time community detection in large networks, *Physical Review E* **79**(6) (2009), 066107.
[29] F. Rosner, A. Hinneburg, M. Röder, M. Nettling and A. Both, Evaluating topic coherence measures, *arXiv preprint arXiv:1403.6397* (2014).
[30] M. Röder, A. Both and A. Hinneburg, Exploring the space of topic coherence measures, in: *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, pp. 399–408.
[31] D. Mimno and D. Blei, Bayesian checking for topic models, in: *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, pp. 227–237.
[32] P. Tijare and P.J. Rani, Exploring popular topic models, in: *Journal of Physics: Conference Series*, Vol. 1706, IOP Publishing, 2020, p. 012171.