IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2023, accepted 12 April 2023, date of publication 24 April 2023, date of current version 2 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3269660

RESEARCH ARTICLE

Detecting Favorite Topics in Computing Scientific Literature via Dynamic Topic Modeling

ROSA VIRGINIA ENCINAS QUILLE[1,2], JOSÉ MELÉNDEZ BARROS[2], MÁRCIO BARBADO JÚNIOR[2], FELIPE VALENCIA DE ALMEIDA[2], AND PEDRO LUIZ PIZZIGATTI CORRÊA[1,2]

[1]School of Arts, Sciences and Humanities, University of São Paulo, São Paulo 03828-000, Brazil
[2]Polytechnic School, University of São Paulo, São Paulo 05508-010, Brazil

Corresponding author: Rosa Virginia Encinas Quille (encinas@usp.br)

This work was supported in part by the São Paulo Research Foundation (FAPESP) under Grant 2019/21693-0; in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 140253/2021-1; in part by the Graduate Program in Information Systems (PPGSI) of the School of Arts, Sciences and Humanities, University of São Paulo; in part by the Graduate Program in Electrical Engineering (PPGEE) of the Polytechnic School, University of São Paulo; and in part by the Institute for Technological Research (IPT).

ABSTRACT Topic modeling comprises a set of machine learning algorithms that allow topics to be extracted from a collection of documents. These algorithms have been widely used in many areas, such as identifying dominant topics in scientific research. However, works addressing such problems focus on identifying static topics, providing snapshots that cannot show how those topics evolve. Aiming to close this gap, in this article, we describe an approach for dynamic article set analysis and classification. This is accomplished by querying open data of notable scientific databases via representational state transfers. After that, we enforce data management practices with a dynamic topic modeling approach on the associated metadata available. As a result, we identify research trends for a given field at specific instants and the referred terminology trends evolution throughout the years. It was possible to detect the associated lexical variation over time in published content, ultimately determining the so-called ''hot topics'' in arbitrary instants and how they correlate.

INDEX TERMS DTM model, data life cycle, text similarity, unsupervised learning, document classification.

I. INTRODUCTION

Text analysis techniques are used in several study areas, mainly in Natural Language Processing (NLP) [1], [2], [3] and text mining [4], [5]. These studies address text classification problems [6] seeking to improve results and generate knowledge. In this context, topic modeling investigations were applied to find ''hot topics'' within the domain of scientific production. As, for example, in previous work, we found ''hot research topics'' using techniques of spectral analysis and text processing [7]. However, the growth and variety of study areas pose significant organizational challenges that require the availability of tools to identify trends and anticipate the appearance of new fields. In this sense, some works have been developed for this purpose [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

Topic modeling tools help detect which areas have higher scientific production and how they evolve. Some works on the extraction and evolution of topics [9] from scientific texts have already been developed using topic modeling to determine the influence and predict future trends of a set of topics in studies on scientific literature [10], [11]. Other works use Latent Dirichlet Allocation (LDA) topic modeling to understand how topics interact over time [12], [13].

In this paper, we use Dynamic Topic Modeling (DTM) algorithm [14] to identify the most relevant topics associated with scientific production in computing science. Besides identifying such topics, we also show how they have evolved in the last 30 years. In addition, the degree of correlation between these topics is calculated, allowing the recognition of similar growth patterns between them.

Approximately one million articles were collected from three different sources: Springer, arXiv, and IEEE Xplore.

The training corpus of the model consists of contents from articles abstracts. The model used here is called Dynamic Topic Modeling, whose hyperparameters are selected after a tuning process with coherence metrics (*c_v* and *c_umass*) and a subsample of 20k articles. Once the hyperparameters of the model were defined, the topics were extracted using the complete sample of documents. Based on the distribution of topics in the documents, the number of documents per topic per year, the correlation between topics and the growth rate were calculated. The probabilities of the words belonging to the topics were then used to determine their evolution over time.

The next sections are organized as follows. Section II describes the related works. Section III defines the data life cycle, resources, and methods used. Section IV describes the tests realized and presents the results. Section V discusses the obtained results. Finally, Section VI presents our conclusions.

II. RELATED WORKS

Dynamic Topic Model was initially proposed by Blei and Lafferty as an extension of the Latent Diritchlet Allocation algorithm [14]. Other LDA-based algorithms have also been proposed to model topic evolution [15]. DTM can capture topics' evolution over time. DTM is applied to cases in which the order of the documents affects the topics, such as the analysis of scientific articles, whose topics depend on the time the article was prepared. This model applies the Markov process to chain the time-specific topic-term distributions under a Logistic-Normal [16], [17]. Blei and Lafferty applied the algorithm to scientific articles that were grouped sequentially by year. They considered the task of predicting articles from a specific year, given the articles from previous years. The results showed a better performance than the LDA since DTM assigned a higher probability to next year's articles than the LDA. Furthermore, it was possible to identify how the distribution of words in topics changed over time. A limitation of the model is that it shows how the initially identified topics change but does not show the appearance or disappearance of new/old topics over time.

Paul and Girju [8] used LDA and DTM to classify papers based on their topics and languages, considering three fields: linguistics, computational linguistics and education. They show how topics vary over time, identify relationships between different fields of study, and analyze trends in scientific production according to the language of origin. Even though this work also employs DTM, our focus is on the whole area of computer science/engineering, aiming to assess the number of articles produced on each topic and the correlation between topics.

Iwata et al. [18] proposed the Multiscale Dynamic Topic Model (MDTM). Unlike the original DTM, MDTM uses non-uniform time intervals, arguing that some words have a longer life cycle than others. The average perplexity results show that the MDTM is far superior to the conventional LDA, and it also slightly exceeds the DTM; however, the computational cost of the MDTM does not offset the gain over conventional DTM.

Yau et al. [19] used LDA and several of its extensions (correlated topic models, Hierarchical LDA and Hierarchical Dirichlet Process) to create clusters of scientific publications; their objective was to explore potential applications in scientometrics. Zhang et al. [20] use DTM to model the time evolution of market competitiveness by capturing and analyzing tweets about different products-services. The work also identifies the topics within that group of products-services (top products) and the brands associated with them, aiming to assess the dominance of the brands over the topics.

Hu et al. [21] applied DTM to identify the evolution of topics in software development. Its objective was to analyze commit messages during the life cycle of a project to capture the strength and evolution of each topic content. Their results showed that DTM could identify more interpretable topics of software evolution. Sleeman et al. [11] used DTM to measure the influence of specific topics on a scientific discipline (namely, climate change), as well as to predict future trends. They used a customized DTM algorithm on a corpus consisting of reports from the Intergovernmental Panel on Climate Change and papers thereby referenced. Then they applied cross-domain analysis to identify correlations between the topics of both corpus and then determined the degree of influence that a given investigation had on a specific report.

Chi et al. [22] used DTM in an Expert Finding system to identify the experts required for a specific field or task. Their work utilizes the method of combining document modelling with profile modeling to carry out the work of Expert Finding. The objective was to identify the topics and keywords within the profiles of the candidates, which are interpreted as associated with their specialities. Lastly, Mihalcea et al. offer a deep approach to knowledge-based measures in [23]. Note that these approaches do not detect favorite topics in computer science literature. In this way, we present a complete approach to analyse and classify sets of dynamic articles in the meta-data in computing scientific literature. We applied data management practices and Dynamic Topic Modeling.

III. METHODOLOGY

This section presents the methodology used to develop this work. The DTM proposal development unfoldings, its associated data life cycle and data visualization plans. In Figure 1, we show the general steps of the methodology.

A. DTM MODEL

Dynamic Topic Modeling is a generative topic model that considers the chronological order of documents where a document is formed of k topics, and each topic is formed by a set of words. The per-document topic distribution α_t and the word distribution $\beta_{t,k}$ of topic k at time t is thus as follows:

1) Draw topics $\beta_{t,k}|\beta_{t-1,k} \sim \mathcal{N}(\beta_{t-1,k}, \sigma^2 I)\forall k$
2) Draw $\alpha_t|\alpha_{t-1} \sim \mathcal{N}(\alpha_{t-1}, \delta^2 I)$

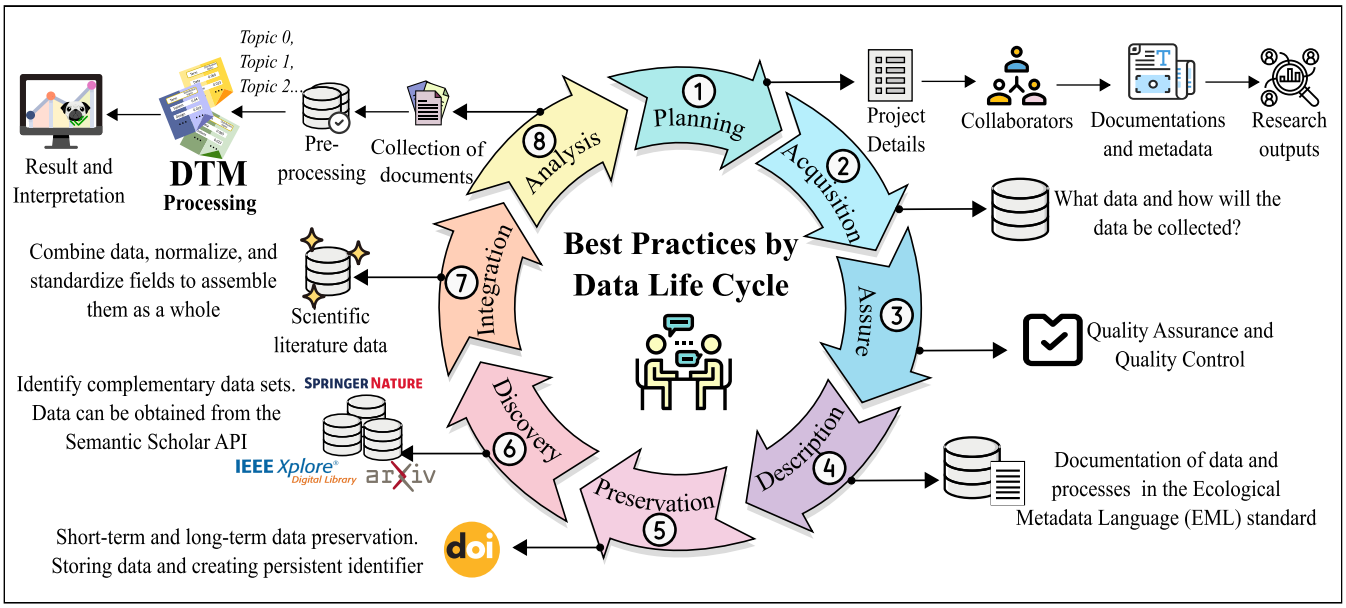

FIGURE 1. Research methodology: Data life cycle and DTM analysis.

3) For each document:
   a) Draw $\eta_{t,d} \sim \mathcal{N}(\alpha_t, a^2 I)$
   b) For each word:
      i) Draw topic $Z_{t,d,n} \sim Mult(\pi(\eta_{t,d}))$
      ii) Draw word $W_{t,d,n} \sim Mult(\pi(\beta_{t,Z_{t,d,n}}))$

where σ, δ and a are variance parameters, $\eta_{t,d}$ and $z_{t,d,n}$ are respectively the topic distribution and the topic for the n word in document d in time t, $w_{t,d,n}$ is a specific word, $Mult(.)$ is the Multinomial distribution and, $\pi(x)$ is a mapping from the natural parameterization x to the mean parameterization, given by Equation 1:

$$\pi(\gamma_i) = \frac{exp(\gamma_i)}{\sum_i exp(\gamma_i)} \tag{1}$$

The number of documents $D_{k,t}$ belonging to topic k in time t is given by Equation 2:

$$D_{k,t} = \sum_{d=1}^{D} \alpha_{k,t,d} \tag{2}$$

where D is the total number of documents.

B. DATA LIFE CYCLE

According to Allard [24], the project known as DataONE understands the data life cycle through eight steps (see Figure 1): (1) planning, (2) acquisition, (3) assure, (4) description, (5) preservation, (6) discovery, (7) integration, and (8) analysis. All those steps were used, and this subsection details each of them.

1) PLANNING

This step is essential for better research management, mapping the processes and resources used throughout the data life cycle. For the planning, documentation was made based on the following items:

- **Project Details:** Specification of the title, objective, summary, financing, and others.
- **Project Contributors:** List of all researchers and data managers.
- **Data Collection:** List of data to be collected and forms of collection.
- **Documents and Metadata:** List documents and metadata accompanying the data.
- **Ethical and legal compliance:** Describe ethical and legal issues treated.
- **Storage, backup, responsibility, and resources:** Describe ways of storage and list of data managers.
- **Selection, preservation, and sharing:** Information on the resources used for selection, preservation, and sharing.
- **Research outputs:** Describe the type of output, expected repositories, and metadata standards.

We define $n \geq 10^6, n \in \mathbb{N}$, as an acceptable quantity of articles to be initially fetched, large enough to meet our work requirements.

2) ACQUISITION

The structure of the data collection process follows the sequence shown in Figure 2 and Figure 3. It is a two-part process: (a) the first part is about querying the research databases, and (b) the second part of this step mainly involves data preprocessing. Our strategy for the second part consists of four steps. The attributes (*id*, *published_unix*, *title*, *abstract* and *author*) that we retain are in a JSON structure. Subsequently, several procedures were applied to clean the data.

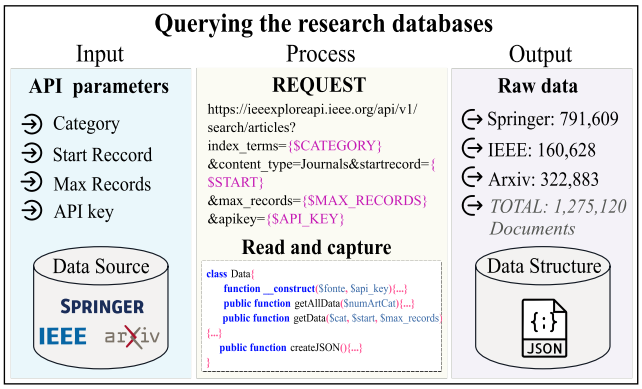

FIGURE 2. First part process for data collection.

Figure 2 summarizes raw data acquisition through a request, read, and capture process with API parameters of the chosen repositories. Note that in this first part, we output "raw data," which is not enough for the analysis process. Consequently, through Figure 3, we present the pre-processing and data cleaning of the "raw data," which is made up of four steps: Step 1, manual removal of attributes that are not significant for the work; Step 2, Data Integration, the process of combining and consolidating data types, taking into account the variety of sources, this variety presents responses in different formats, structure conflicts, and semantic conflicts; Step 3, data cleaning, which makes up a set of tasks. Finally, Step 4 is data transformation to convert raw data from one format to another that contributes to the work.

Data cleaning removes duplicity, empty data, line breaks, little information, a language other than English, punctuation marks, alphanumeric symbols, numbers, and numeric symbols. Next, natural processing techniques are used: Tokenization as the first step [25], [26]; removal of stopwords, words with high frequency and without significant meaning help in the interpretation process [27], [28]; and Lemmatization groups the forms of a word so that they can be analyzed as a single element [29].

After the process with PLN, it was necessary to insert inclusion and exclusion criteria:

1) Exclusion of infrequent words representing a long tail in the word frequency graph.
2) Exclusion of short texts that were left with few tokens due to previous data cleansing steps.
3) Inclusion of articles from 1990 onward.

In Figure 4, we present an example of titles that need to be pre-processed. We can observe that symbols, signs, and words are irrelevant to this text analysis.

3) ASSURE

At this step, we used techniques that helped to improve data quality [30]. This process was divided into two stages. In stage one, the following tasks were performed:

- Remove Useless Attributes.
- Elimination of repeated articles.

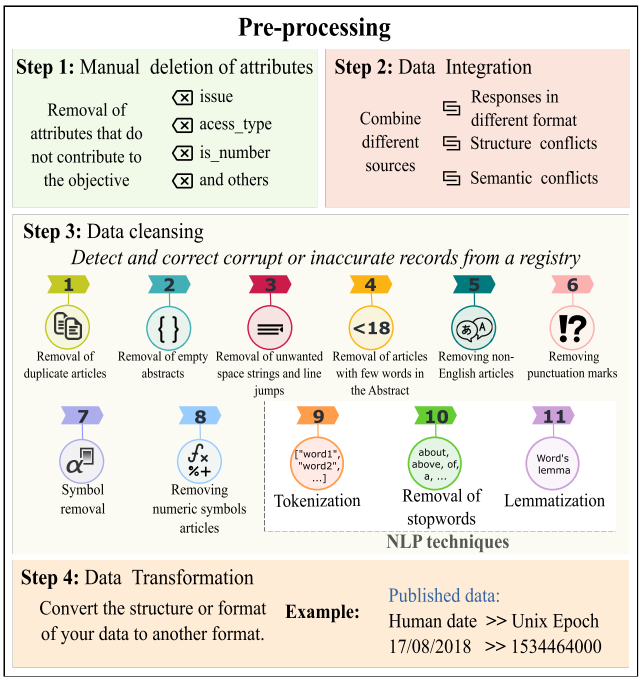

FIGURE 3. Second part process for data collection.

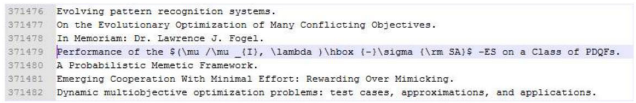

FIGURE 4. Example of article titles before pre-processing.

- Removal of articles whose abstracts had inconsistent content.
- Deletion of blank space chains and unwanted line breaks.
- Articles with abstracts of less than 18 words and whose language was not English were excluded.

Stage two was developed using the *NLTK* and *Spacy* libraries in python:

- The raw text data were tokenized (implemented with *nltk.tokenize.word_tokenize*).
- Trivial and common words that generally do not add much detail to the meaning of a body of text (Stop words) were removed (the stop words were set using *nltk.corpus.stopwords*).
- Lemmatization was applied to return to the base or dictionary form of each word (implemented with *spacy.lemmatizer*). Only verbs, nouns and adjectives were kept.
- Infrequent words were deleted (abs. frequency < 70).
- Finally, as the DTM does not work properly with short texts, those abstracts with less than 40 tokens were removed. Only articles dated from 1990 are kept.

4) DESCRIPTION

The Ecological Metadata Language standard and the Morpho tool were used for the metadata documentation.

5) PRESERVATION

A version manager was used for short-term data preservation, long-term open data repositories, and the creation of persistent identifiers. Data publishing, including creating a Digital Object Identifier (DOI) for the generated dataset in the IEEE DataPort. And the publication of the code, including a DOI on the Code Ocean platform. We make it available in the following links: DataPort https://dx.doi.org/10.21227/7exb-wb55 and Code Ocean https://doi.org/10.24433/CO.1245883.v1.

6) DISCOVERY

Some potentially useful data, in addition to those already mentioned, may be the references used and the citations made for each. These data would allow establishing connections between articles and obtaining a flow through the topics to improve the results of the forecasts. These data can be obtained from the Semantic Scholar API.

7) INTEGRATION

As we used three different sources, there was also the need for normalization and standardization of fields for assembling them as a whole. Each source used has its own structure and attributes. Structures conflicted in many ways, e.g., the hierarchy through which authors are attached to the formal structures of returned entries, the format in which the data was returned, and semantic inconsistencies. In Section IV, we detail how the integration process was developed to solve each of these inconsistencies.

8) ANALYSIS

Cluster analysis – for analyzing the clustering, Dynamic Topic Modeling has been used to identify clustering within texts (hot topics) and the quantities of articles associated with each topic each year. Coherence measures were used to identify the best cluster partition. The Pearson correlation was used to identify if there was any relationship between the growth rate of some topics.

C. DATA VISUALIZATION

For the purposes of this work, the *matplotlib* library was used to visualize the number of documents associated with each topic over time, the growth rate and its evolution.

IV. TESTS AND RESULTS

Software tests were conducted on the following databases: the arXiv Database, IEEE Xplore Digital Library, and Springer Nature Metadata. Each has its representational state transfer (REST) Application Programming Interface (API) library for metadata extraction via HTTP GET requests.

Queries to the arXiv database must be made according to the following structure [31]:

- http://export.arxiv.org/api/query?search_query=cat:CATEGORY&start=START&max_results=MAX_RESULTS

TABLE 1. Articles per source.

Source	Works	Works (%)	Size (MB)	Size(%)
arXiv	322,883	25.32	640	49.84
IEEE Xplore	160,628	12.60	156	12.15
Springer Nature	791,609	62.08	488	38.01
Total	1,275,120	100.00	1,284	100.00

Parameter *cat* determines the category, parameter *start* can be set with the registry from which to start, and parameter *max_results* stands for the number of results to be queried (the API returns 3000 XML-formatted results per query).

IEEE Xplore Digital Library service [32] requests are structured as:

- https://ieeexploreapi.ieee.org/api/v1/search/articles?index_terms=CATEGORY&content_type =Journals&startrecord=START&max_records=200&apikey=API_KEY

The *index_terms* variable contains the categories to use, and *content_type* specifies the publication type. Only articles published in IEEE journals were considered for this work. Variable *start_record* specifies a starting point, and *max_records* specifies the maximum number of results retrieved per query (the maximum allowed is 200). The *apikey* variable contains the developer key, provided by [33].

Springer Nature service [34] offers four interface types. This work uses the one named API Springer Nature Meta, which is based on requests with the following structure:

- http://api.springernature.com/meta/v2/json?q=subject:%22Computer%20Science%22&s=START&p=100&api_key=API_KEY

The value after *subject* defines the area (in this case, Computer Science), the *s* variable receives the starting point value, *p* receives the number of results per query (the maximum allowed is 100), the *api_key* variable receives the specific developer key, obtained through registration in [34].

Regarding programming languages, we have used PHP to automate queries and Python to implement DTM. The PHP script saves data locally in JSON format. The raw data obtained can be seen in Table 1.

Once the raw data were obtained, cleaning these data was applied (See section III-B3).

At the end of the cleansing process, there were 939,452 articles left, that is, 73.67% of the initial number. The computation time of the cleaning process was approximately 142 minutes in Google Colab.

Regarding data integration, although the collected objects correspond to the same class, each API has its structure, format, and attributes. The main differences found were the following:

- arXiv API delivers the data in XML format, while Springer and IEEExploran delivered it in JSON and optionally in XML. Since we chose JSON to handle our data structure, it was necessary to convert the arXiv XML files.

- The publication date formats were also a problem because each database works with a different format. We converted them all to integers in the Unix epoch format, which has turned variable manipulation into less complex arithmetic operations.
- Structure conflict: The hierarchies between objects and attributes have different structures across objects of the same type, as in the case of the authors. Springer's authors are member objects of a ''creators'' attribute. While in IEEExplore, the authors are objects that belong to an ''authors'' attribute, which in turn belong to an ''authors'' object.
- Semantic conflict: The same attribute could be described with a different label, as in the case of ''abstract'' and ''summary'' or ''author'' and ''creator.''

To resolve all those inconsistencies, the results gathered were organized according to the JSON structure. Later, we used a JSON Schema Validator to verify that the objects complied with the structure required for the project.

After fetching from scientific databases and filtering, a tuning process was applied to identify the most suitable hyperparameters for the DTM model. The model was implemented using the Gensim library [35] and its corresponding repository [36]. The process was developed using a random subset of 18,993 papers, 20-time slices, $0.01 \leq \theta \leq 0.11$ and $4 \leq k \leq 10$. In order to expedite the tuning process, θ was adjusted in intervals of 0.02. In total, 42 partitions were generated, and the results of each were validated using the coherence metrics c_v and c_umass. Figure 5 (a and c) shows the average values of each metric per k-topic. The greater the number of k-topics, the distribution of words in documents tends to be more homogeneous; that is, the higher the k, the higher the performance. Therefore, it is not objective to directly compare metrics between different values of k-topics; instead, we used the variation rate by calculating the ratio between $[M(k-1) - M(k)]$ and $[M(k) - M(k+1)]$ (Figure 5b and 5d).

Considering the results, the best options were k-topics $=$ 9 and k-topics $= 7$. The θ values corresponding to the best results of $k = 9$ and $k = 7$ were 0.05 and 0.01, respectively. Afterward, the two final partitions ($k = 7$ and $k = 9$) were generated using the entire document set and 30 time slices. Based on the interpretability of the words associated with each topic, we determined that $k = 7$ was the set that best represented the description of the topics. The seven topics are presented in Table 2. The obtained number of articles per topic is presented in Table 3. All of the data from this work are available at IEEE DataPort [37].

We use a 20-time and then 30-time slice to get an idea of which hyperparameters could best help to have good accuracy since the model takes a long time to complete the execution. In other words, we select a small sample, and after identifying the best hyperparameters, we repeat the experiment with the complete data.

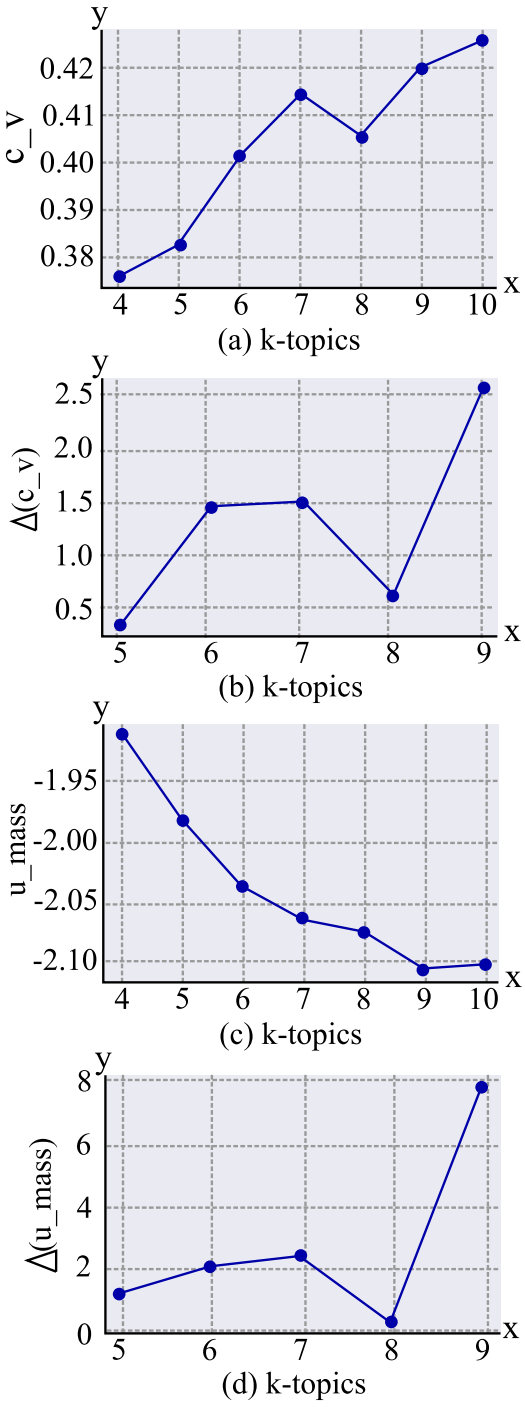

FIGURE 5. Coherence metrics c_v and c_umass for every k-topic.

In Table 3, column *Topic* stands for the subject sets translated and listed in Table 2. Column *Quantity* shows the obtained number of articles for a given Topic, and in %, the corresponding percentage. The last two columns portray each topic's average production per year, first between 2010 and 2014 and then between 2015 and 2019. The last line presents total values. Figure 6 presents a smooth scatter plot for the number of published articles on each topic through recent

TABLE 2. Top seven topics.

Code	Subject(s)
Topic 0	Security, access control, encryption scheme, cryptography, cloud computing
Topic 1	Computational complexity, optimization, graph partition, numerical analysis, combinatorics
Topic 2	Data-exchange, semantics
Topic 3	Computer vision, object detection, recognition, video tracking
Topic 4	Interactive computing, human-computer interaction, human-centered design, business management systems
Topic 5	Artificial intelligence, machine learning, clustering
Topic 6	Energy efficiency, power systems, network communication

TABLE 3. Number of articles per topic.

Topic	Quantity	%	Avg. 2010-14	Avg. 2015-19
Topic 0	80833	8.60%	4113.78	6581.21
Topic 1	14630	15.57%	7712.39	12091.07
Topic 2	113134	12.04%	5873.32	7305.74
Topic 3	129591	13.79%	6398.86	10729.49
Topic 4	191219	20.35%	10379.87	13985.50
Topic 5	153695	16.36%	6859.11	14876.22
Topic 6	124678	13.27%	6486.87	10404.38
	939452	100%	6832.03	10853.37

years. Their growth rates over recent years are presented in their totalities in Figure 7, and source dissociated in Figure 8. Figure 9 shows a sample of the results delivered by the DTM algorithm regarding topic five words. Evolution over time for the words making up the such topic is represented by changes in their probabilities of being associated with the topic.

V. DISCUSSION

Considering the data collected from specific scientific databases, the DTM modeling makes it possible to point out scientific research's topmost tendencies for specific instants in time and associated lexical variations through the years.

According to the results portrayed in Figure 6, all the topics grew in publication numbers from 1990, which is compatible with recent reports [38]. Great remarks can be made about Topic 4, mainly related to human-computer interfacing, and Topic 5, mostly related to artificial intelligence. The former went well as the most published subject from circa 1997 up until recently, when it lost momentum, being surpassed by a growing number of publications in Topic 5, which has been rising steadily after 2015 up to the current days. Evidence of this is that the percentage increase in the number of articles in 2015 compared to 2011 was 28.24%, while in 2019, compared to 2015, it was 173.16%. Table 3 shows how topic 5, in those same periods, went from producing 6,859 to 14,876 articles per year. Remarkably, this finding of Topic 5 is confirmed by a recent work by Faraboschi et al. [39], in which predictions about deep learning technologies receive an A score in the IEEE scorecard. Topic 5, Topic 4 and Topic 1, in this descending order, are currently the three most published subjects, according to our findings. In general, 5 of the seven topics identified show a steep inclination in the article production curve in the last three years, which shows of how prolific the area has become in recent times.

Figure 7, shows the growth rate of one year with respect to the previous year. A noteworthy fact is that from 2002-2019, the pattern of growth rates of the topics seems to be more homogeneous compared to 1990-2001 when there was more significant heterogeneity. This is evidenced when calculating the mean positive correlation between topics in these two periods; the results are 0.71 (2002 - 2019) and 0.45 (1990 - 2001). From 2002-2019, there is no negative correlation between topics; in 1990-2001, the average negative correlation between topics is -0.31.

Another aspect to highlight is the period 2005 - 2009. The topics significantly grew from 2001 to 2005, but production plummeted in 2006. In 2007, there was a slight recovery, and a further drop in 2008. This may be due to a fact linked to one of the data sources and has not been generalized because the number of documents taken from the three sources was not balanced. Figure 8 shows the growth rate of each source separately. Although it is perceived that the Springer graph is the dominant one, both Springer and IEEExplorer show that, indeed, there was a slowdown in the 2005–2006 period; in arXiv, this situation is only partial. The three sources also coincide with the acceleration of the 2006–2007 period. Outside those years, Springer seems to dominate production, so specific variations in growth can be linked to an event of the source itself and not be an inherent fact of the studied object.

From Figure 9, it is possible to observe that the words *neural* and *network* have suffered a decline to levels of 0.5% from the early 1990s until 2008, and after that, both tended to increase. An aspect in Figure 9 regards the word *deep*, which is usually associated with the *deep learning* term. It turns out that *deep* debuts only in 2013, and from then on, its occurrence has increased. We validated this behavior in sources other than those used in this research. For example, in ACM Digital library, considering the 13,156 entries with the *deep learning* expression, 97.9% pointed to the 2013–2020 period. On Science Direct, the percentage is 95.8% (25,161 of 26,264), and on Semantic Scholar, it is 97.1% (206,000 of 212,000).

Figures 10 to 16 show the evolution of the ten most relevant words found in each topic with their probabilities between the years 1990 to 2019.

In Topic 0, shown in Figure 10, the words that grew over the years are *attack* and *security*, ranking first and second as the most relevant words, showing a strong relationship with the topic. In addition, other words appeared, such as *privacy* in 2003 and *cloud* in 2011. These words are interesting since studies on security have been growing in the areas of Cloud Computing (for example, privacy in public Cloud Computing), Encryption (for example, public and private keys), and Access Control.

In Topic 1, illustrated in Figure 11, the words *problem* and *algorithm* lead despite showing a slight drop in recent

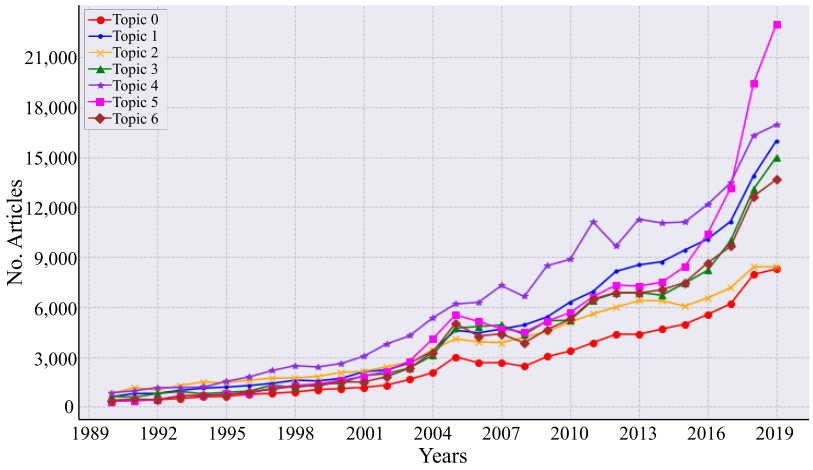

FIGURE 6. Number of published articles evolution for each topic.

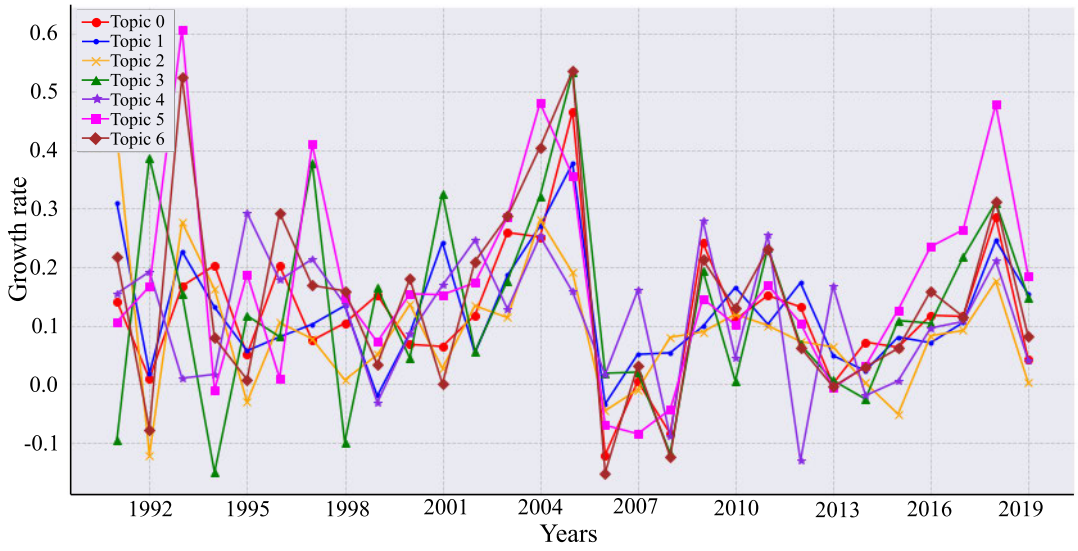

FIGURE 7. Total growth rate of the number of topic 0–6 articles through the years.

years. This topic maintains its most representative words over time; this indicates that the topic prevails in importance. For example, based on history, studies have been related to computational complexity theory since the 1960s [40]. Another example would be the Turing machines that are based on this topic and that, in turn, had great repercussions in the history of Computer Science.

In Topic 2 (see Figure 12), the evolution of the representative words tends to increase and others to decrease, as can be observed in words *system*, *model*, and *process*, which tend to grow. On the other hand, the words *language*, *object*, *logic*, and *program* tend to go down. This can be interpreted as a study approach more focused on systems, models, and processes. Note also that from 2002 the word *ontology* expanded because this concept in this topic is very required, for example, in semantic studies.

In Topic 3 (Figure 13), the evolution of words is very similar to Topic 1, maintaining some of its most representative words over time. Only the word "image" tends to grow strongly, and the word "video" appears in 1995 and also tends to grow, which makes sense within the topic.

In Topic 4 (Figure 14), the representative words vary in time. The word "user" can be considered a keyword within the topic since it is user-focused, such as human-computer interaction, UX, or UI. We also see the words *web*, *service*, and *social* appearing in more recent years; these terms also focus on the user in different contexts.

In Topic 5 (Figure 15), the evolution of the words is clear. We can notice three interesting aspects: (1) the fall and rise of the words *network*, *learn*, and *neural* over time; (2) The words *cluster*, *pattern* and *training* seem to remain; and finally (3) the appearance of the word *deep* with a strong tendency to

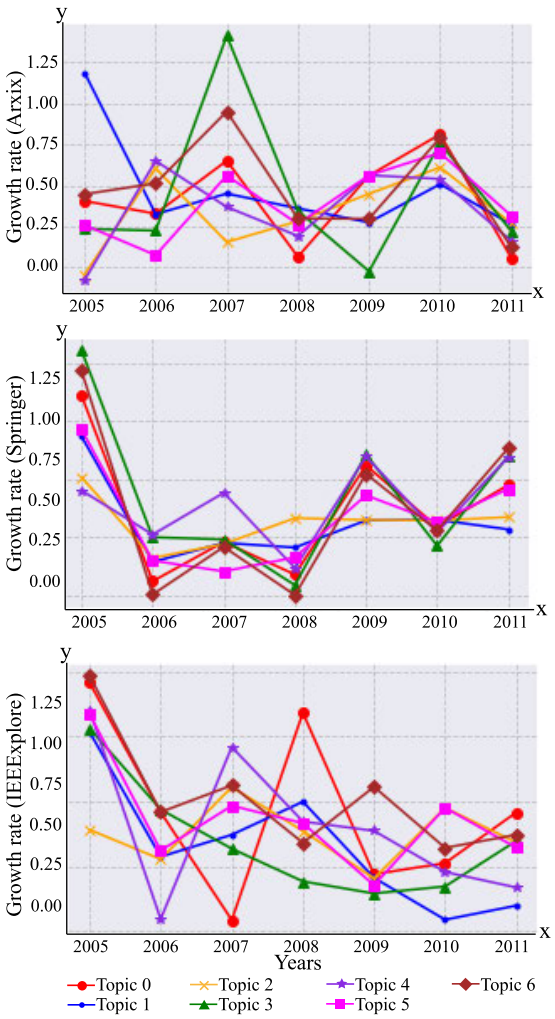

FIGURE 8. Growth rate for each source.

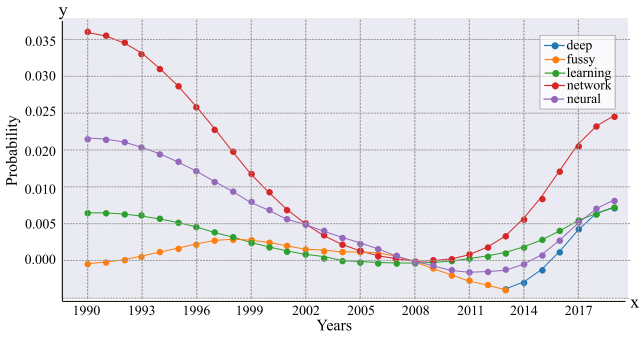

FIGURE 9. DTM evolution of topic 5 words: artificial intelligence, machine learning, and clustering.

grow. This last word refers to studies with Deep Learning, one of the most current topics within Artificial Intelligence.

For Topic 6 (Figure 16), the representative words vary over time, but they are very evident in each era, such as the appearance of the words *wireless*, *sensor* and *energy*.

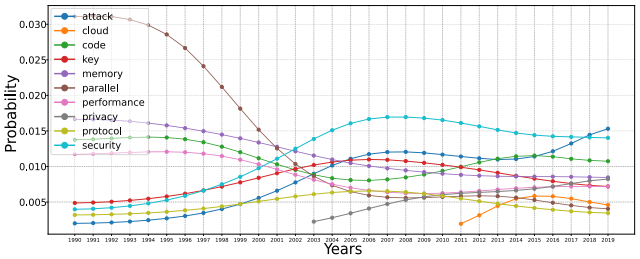

FIGURE 10. DTM evolution of the ten most relevant words for Topic 0 between 1990-2019.

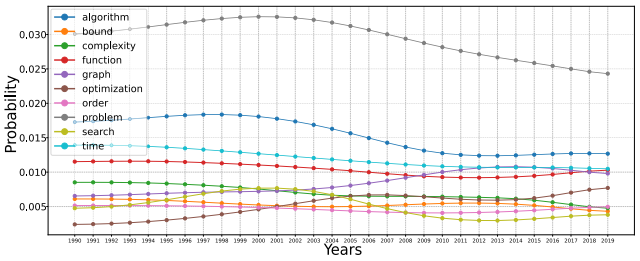

FIGURE 11. DTM evolution of the ten most relevant words for Topic 1 between 1990-2019.

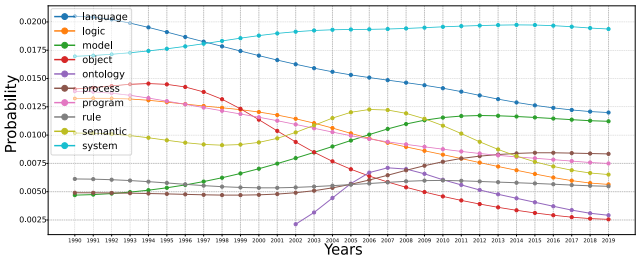

FIGURE 12. DTM evolution of the ten most relevant words for Topic 2 between 1990-2019.

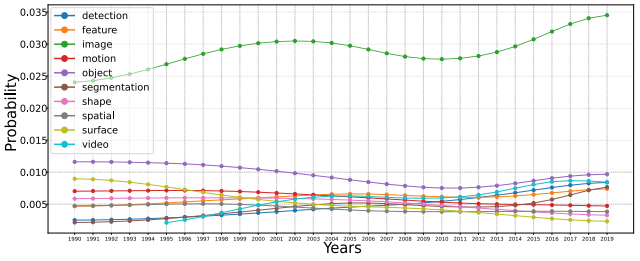

FIGURE 13. DTM evolution of the ten most relevant words for Topic 3 between 1990-2019.

This work uses a sample of the collected data. Thus, considering the whole and available resources at the research time, an estimated total of 480 tuning hours would be necessary for choosing appropriate hyperparameters. A recommendation for future work would be to implement the tuning and cleaning process in cloud computing infrastructures, using Apache Spark and the Cassandra database to parallelize the process and reduce computation time.

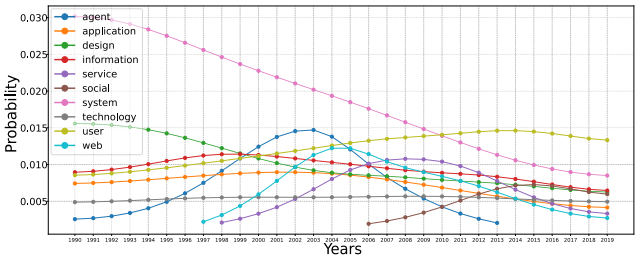

FIGURE 14. DTM evolution of the ten most relevant words for Topic 4 between 1990-2019.

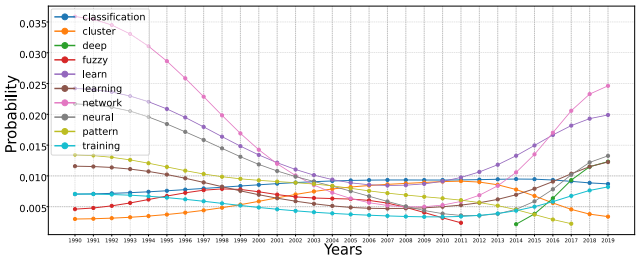

FIGURE 15. DTM evolution of the ten most relevant words for Topic 5 between 1990-2019.

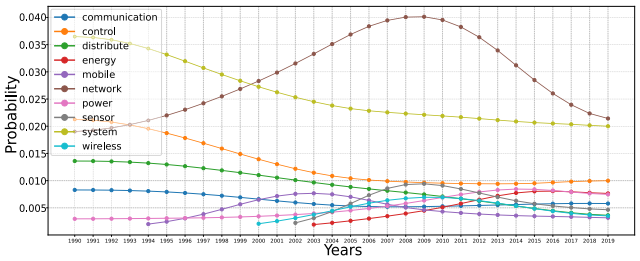

FIGURE 16. DTM evolution of the ten most relevant words for Topic 6 between 1990-2019.

For those interested in exploring the other topics in detail, all code is available on IEEE Code Ocean [41]. It allows generating different graphs by manipulating the topic variables, years and words. Regarding code, simple adaptations can provide specific information, e.g., top authors for a given topic. Moreover, a hypothesis that could be worked on is whether the similarity between the growth rate patterns in the last years is because cooperation between different areas has increased compared to previous periods.

Additionally, something to be further investigated is the 2008 word convergence.

VI. CONCLUSION

In this work, we presented a dynamic article set analysis and classification process. This proposal uses a set of data management steps with a Dynamic Topic Modeling approach on the associated metadata available. We address the problem of identifying dominant topics in scientific research in a dynamic way and how they evolve. We performed experiments on data sets from the ArXiv Database, IEEE Xplore Digital Library, and Springer Nature Metadata to demonstrate that using text analysis in Dynamic Topic Modeling can identify research trends for a given field at specific instants. Also, it can detect the associated lexical variation over time in published documents, determining "hot topics" in arbitrary instants and how these correlate with each other; our results showed that this is possible.

An important future work is to expand the possibilities of using Bigrams/n-grams within NLP. The Bigram Language Model is a word formation process. For example, in Artificial Intelligence, it is common to find words such as "Bayesian Network" and "Neural Network." Where both are composed by "Network." In our NLP, we obtained three words (bayesian, neural, and network), whereas, with Bigrams, we would have two words (bayesian-network and neural-network). The use of Bigrams could enable more specific results in terms of the top words of each topic, expanding the capabilities of our topic detection methodology via DTM. A recommendation for another future work is to carry out experiments at different levels of abstraction. It can be for a research area/sub-area/theme; for example, finding AI hot-topics, or environmental study hot-topics, among others.

ACKNOWLEDGMENT

The authors would like to thank Maria Cristina Vidal Borba for her careful revision.

REFERENCES

[1] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, Jul. 2015.
[2] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *AI Open*, vol. 3, pp. 71–90, 2022.
[3] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023.
[4] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," *J. Comput. Linguistics Lang. Technol.*, vol. 20, no. 1, pp. 19–62, 2005.
[5] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text mining in big data analytics," *Big Data Cognit. Comput.*, vol. 4, no. 1, p. 1, Jan. 2020.
[6] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.
[7] R. V. E. Quille, C. Traina, and J. F. Rodrigues, "Spectral analysis and text processing over the computer science literature: Patterns and discoveries," in *Proc. 29th Annu. ACM Symp. Appl. Comput.*, New York, NY, USA, Mar. 2014, pp. 653–657.
[8] M. Paul and R. Girju, "Topic modeling of research fields: An interdisciplinary perspective," in *Proc. Int. Conf. RANLP*, Borovets, Bulgaria, Sep. 2009, pp. 337–342.
[9] Z. Li, Z. Yin, and Q. Li, "Study on topic intensity evolution law of web news topic based on topic content evolution," in *Proc. 4th Int. Conf. Cloud Comput. Secur.* Cham, Switzerland: Springer, 2018, pp. 697–709.
[10] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles, "Detecting topic evolution in scientific literature: How can citations help?" in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, New York, NY, USA, Nov. 2009, pp. 957–966.
[11] J. Sleeman, M. Halem, T. Finin, and M. Cane, "Discovering scientific influence using cross-domain dynamic topic modeling," in *Proc. IEEE Int. Conf. Big Data*. Boston, MA, USA, Dec. 2017, pp. 1325–1332.
[12] X. Wang and A. McCallum, "Topics over time: A non-Markov continuous-time model of topical trends," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2006, pp. 424–433.
[13] D. Duarte, L. H. Rocha, and D. Welter, "What happened in 2020: A topic modeling approach based on a topic similarity metric," *Brazilian J. Inf. Syst.*, vol. 15, pp. 19:1–19:17, Oct. 2022.

[14] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. 23rd Int. Conf. Mach. Learn.*, New York, NY, USA, 2006, pp. 113–120.
[15] A. T. Wilson and D. G. Robinson, "Tracking topic birth and death in LDA," Sandia Nat. Lab., Albuquerque, NM, USA, Tech. Rep. SAND2011-6927 and TRN: US201201%%236, 2011. [Online]. Available: https://www.osti.gov/biblio/1029827
[16] M. Emoto, "Method for extraction of purchase behavior and product character using dynamic topic model," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 778–782.
[17] A. Bhadury, J. Chen, J. Zhu, and S. Liu, "Scaling up dynamic topic models," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 381–390.
[18] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda, "Online multiscale dynamic topic models," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2010, pp. 663–672.
[19] C.-K. Yau, A. Porter, N. Newman, and A. Suominen, "Clustering scientific documents with topic modeling," *Scientometrics*, vol. 100, no. 3, pp. 767–786, Sep. 2014.
[20] H. Zhang, G. Kim, and E. P. Xing, "Dynamic topic modeling for monitoring market competition from online text and image data," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2015, pp. 1425–1434.
[21] J. Hu, X. Sun, D. Lo, and B. Li, "Modeling the evolution of development topics using dynamic topic models," in *Proc. IEEE 22nd Int. Conf. Softw. Anal., Evol., Reeng. (SANER)*, Mar. 2015, pp. 3–12.
[22] R. Chi, B. Wu, and L. Wang, "Expert identification based on dynamic LDA topic model," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2018, pp. 881–888.
[23] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proc. AAAI*, vol. 1, 2006, pp. 775–780.
[24] S. Allard, "DataONE: Facilitating eScience through collaboration," *J. eSci. Librarianship*, vol. 1, no. 1, pp. 4–17, 2012.
[25] J. J. Webster and C. Kit, "Tokenization as the initial phase in NLP," in *Proc. 14th Conf. Comput. Linguistics*, 1992, pp. 1–5.
[26] C. Fox, *Lexical Analysis and Stoplists*. Upper Saddle River, NJ, USA: Prentice-Hall, 1992, pp. 102–130.
[27] A. Blanchard, "Understanding and customizing stopword lists for enhanced patent mapping," *World Pat. Inf.*, vol. 29, no. 4, pp. 308–316, Dec. 2007.
[28] R. T.-W. Lo, B. He, and I. Ounis, "Automatically building a stopword list for an information retrieval system," in *Proc. 5th Dutch-Belgian Inf. Retr. Workshop*, vol. 5, 2005, pp. 17–24.
[29] T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola, "Stemming and lemmatization in the clustering of Finnish text documents," in *Proc. 13th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Nov. 2004, pp. 625–633.
[30] M. Nesca, A. Katz, C. Leung, and L. Lix, "A scoping review of preprocessing methods for unstructured text data to assess data quality," *Int. J. Population Data Sci.*, vol. 7, no. 1, pp. 1–10, Oct. 2022.
[31] (2019). *arXiv API*. Accessed: Dec. 2, 2019. [Online]. Available: https://arxiv.org/help/api
[32] IEEE Xplore. (2019). *IEEE Xplore API Portal*. Accessed: Dec. 2, 2019. [Online]. Available: https://developer.ieee.org/
[33] IEEE Xplore. (2019). *Register for an Account*. Accessed: Dec. 3, 2019. [Online]. Available: https://developer.ieee.org/member/register
[34] S. API. (2019). *Springer Nature API Portal*. Springer. Accessed: Dec. 2, 2019. [Online]. Available: https://dev.springernature.com/
[35] R. Technologies. (2019). *Dynamic Topic Models (DTM) and the Document Influence Model (DIM)*. [Online]. Available: https://radimrehurek.com/gensim/models/wrappers/dtmmodel.html
[36] D. M. Blei and J. D. Lafferty. (2015). *Dynamic Topic Models and the Document Influence Model*. [Online]. Available: https://github.com/blei-lab/dtm
[37] J. M. Barros, R. V. E. Quille, M. B. Júnior, P. L. P. Corrêa, G. de Bona, and M. A. Simplicio Jr., "Topics modeling in computer science articles," Inst. Elect. Electron. Eng., New York, NY, USA, 2020.
[38] E. Landhuis, "Information overload," *Nature*, vol. 535, pp. 457–458, Jul. 2016. [Online]. Available: https://www.nature.com/articles/nj7612-457a.pdf
[39] P. Faraboschi, E. Frachtenberg, P. Laplante, K. Mansfield, and D. Milojicic, "Technology predictions: Art, science, and fashion," *Computer*, vol. 52, no. 12, pp. 34–38, Dec. 2019.
[40] D.-Z. Du and K.-I. Ko, *Theory of Computational Complexity*, vol. 58. Hoboken, NJ, USA: Wiley, 2011.
[41] J. Barros, R. Quille, M. Júnior, F. Almeida, and P. Corrêa, "Dynamic topic modeling [source code]," Code Ocean-North America HQ, New York, NY, USA, 2020.

ROSA VIRGINIA ENCINAS QUILLE received the degree in systems engineering from the National University of the Altiplano, Peru, in 2007, and the master's degree in computer science and computational mathematics from the University of São Paulo (USP), Brazil, in 2014. She is currently pursuing the dual Ph.D. degree with the Information System of EACH, USP, and the EPUSP Big Data Research and Extension Group.

JOSÉ MELÉNDEZ BARROS received the bachelor's degree in industrial engineering from the University of La Guajira, Colombia, in 2012. He is currently pursuing the M.Sc. degree with the Computer Engineering and Digital Systems Department (PCS), Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil. His research interests include machine learning, natural language processing, and web technologies.

MÁRCIO BARBADO JÚNIOR received the bachelor's degree in computer science from Universidade Paulista, in 2017. He is currently pursuing the M.Sc. degree with the Computer Engineering and Digital Systems Department (PCS), Escola Politécnica da Universidade de São Paulo (Poli-USP). He was with software development for many years in the private sector, where he dealt mainly with information security projects, the same field of his current research, conducted with the LARC Laboratory, Poli-USP, and focused in lattice-based cryptography.

FELIPE VALENCIA DE ALMEIDA received the degree in computer engineering and the master's degree in electrical engineering—concentration area computer engineering from the University of São Paulo, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with a focus on research in the area of high performance programming and emphasis on the application of accelerators implemented in FPGAs and heterogeneous architectures.

PEDRO LUIZ PIZZIGATTI CORRÊA received the bachelor's and master's degrees in computer science from Universidade de São Paulo, São Paulo, Brazil, in 1987 and 1992, respectively, and the Ph.D. degree in electrical engineering from Escola Politécnica, University of São Paulo, in 2002. He was a Postdoctoral Researcher in data science with the University of Tennessee, in 2015, and an Associate Professor with the University of São Paulo, in 2017, where he is currently an Associate Professor with the Computer Engineering and Digital Systems Department (PCS), Escola Politécnica, working mainly on distributed databases, data science, the modeling of computer systems, the architecture of distributed systems, computing and biodiversity, agricultural automation, and electronic government.