Chapter 6
Burst Detection

Abstract This chapter provides a theoretical framework for burst detection, including its advantages, disadvantages, and other essential features. It further enumerates various open-source tools that can be used to conduct burst detection and discusses the use cases on how the information professionals can apply it in their daily lives. The chapter is followed by a case study using two different tools to demonstrate the application of burst detection in libraries.

6.1 What Is Burst Detection?

A time series can be defined as a sequence of events or observations over time. A time-series data can be discrete or continuous, for instance, Wikipedia editing activity or Google Trends[1] showing results for Google searches for different queries. Time-series events are commonly represented by lists where a record represents each event with temporal attribute(s) (Fig. 6.1). The following are some of the common patterns in a time-series data:

1. Trends: It corresponds to the general tendency of time-series data to be increasing, decreasing, having stability, or be in a cyclic tendency.
2. Seasonality: These are the repetitive and predictable movement around a trend line such as cyclic variations of flu infections or harvesting of crops.
3. Bursts: These correspond to the identification of sudden spurt of activity, sometimes in response to an external event such as a disease.

Burst detection is a temporal analysis that "aims to identify the nature of phenomena represented by a sequence of observations such as patterns, trends, seasonality, outliers, and bursts of activity" [1]. It can be used to (i) understand the temporal distance such as the most emerging or trending terms, growth of terms, and latency/peak or (ii) forecasting, that is, predicting, future values of the time-series variables. Various algorithms [2–6] perform burst detection, but Kleinberg's

[1] https://trends.google.com/.

M. Lamba, M. Madhusudhan, *Text Mining for Information Professionals*,
https://doi.org/10.1007/978-3-030-85085-2_6

(a)

Subject header	Date	Time
Meeting for project	1/1/2021	10:03 AM
Invitation to give talk	1/1/2021	10:05 AM
Review paper	1/1/2021	12:01 PM
Happy New Year!	1/1/2021	12:01 PM
To do list reminder	1/1/2021	12:04 PM

(b)

Month	Day	Year	Time	Rank
12	09	2021	20:00	75,000
12	09	2021	22:00	65,975
12	09	2021	23:00	70,313
12	10	2021	2:00	98,209
12	10	2021	3:00	106,957

Fig. 6.1 Examples showing temporal data. (**a**) Emails. (**b**) Amazon books' rankings

algorithm [7] is the most popular one that detects bursts in discrete batches of events. This algorithm identifies the bursts of topics for a temporal stream of documents based on the Hidden Markov Model. "Originally developed to detect topics in email chains, Kleinberg's method assumes that terms in documents are emitted by a two-state automaton. The automaton may spontaneously transit from a non-bursty state to a bursty state, or vice versa" [8]. It is "commonly applied to identify words that have experienced a sudden change and increase in the frequency of occurrence. The algorithm generates a list of word bursts in the document stream, ranked according to the burst weight, together with the intervals of time in which these bursts occurred. The burst weight depicts the intensity of the burst, i.e., how great the change in the word frequency that triggered the burst" [1]. Thus, burst detection helps in the classification and detection of keywords and in "maintaining a record of the event based on related features" [5]. These features help to find the trending pattern of events meticulously at different times.

6.1.1 How to Detect a Burst?

Textual data can be imagined as a discrete time series where a sequence of observations/events occurs over time. These observations happen at regularly spaced intervals such as every month, week, volume, issue, or year. *Bursts* are the large number of events taking place within a defined period. In contrast, *burstiness* is defined as the events concentrated within a short period of intense activity followed by long intervals of inactivity. An event can be planned or unplanned and can be trending or non-trending, which is directly associated with the event's frequency. For instance, planned events such as IPL 2021 Cricket is an example of bursty event, whereas American Idol Premier is also a planned event but non-bursty. "Similarly, unplanned events such as Cyclone 2021, India was bursty, and a minor road accident is non-bursty. The bursty behavior is directly associated with the diffusion of information over social media or a network" [5]. There are two approaches to detect a burst:

1. **Analyzing complete data stream**: This approach is not appropriate for identifying bursts for every occurrence of an event.
2. **Using multiple time windows**: This approach can identify bursts in real time by decreasing window size and calculates the combined occurrence of events within the time window. Moreover, this approach helps in storing the collection of every event for each period.

Real-time detection of a burst at an early stage in emails, news, blogs, research articles, tweets, comments, and Internet bulletin boards is becoming gradually significant in several fields where topics appear and grow in intensity rapidly. To detect a burst and report it as early as possible is quite valuable for essential decision-making processes within a discipline. Various methods, like content analysis [9], clustering [10–16], search [17], and personalization [18], have been proposed for burst detection.

6.1.2 Comparison of Burst Detection with Others

Topic Detection and Tracking (TDT)[2] was first used to solve the issue of tracking topics for time-ordered problems, but with the advancement in computer science research, Latent Dirichlet Allocation (LDA)[3](see Chap. 4) became the most popular approach for topic modeling. However, topic modeling has many disadvantages, for instance, "the lack of interpretability of the results and the difficulty in coherently linking LDA topics together between subsequent time-steps" [8]. In contrast to LDA, burst detections "first identify the bursty terms in a dataset, and then cluster them together into topics" [8]. Singh and Kumari [5] compared their burst detection technique (*BUrST*) with other state-of-the-art methods like LDA, graph-based feature-pivot topic discovery, document-pivot topic detection, and BN-grams method. They found *BUrST* to be proficient in detecting valuable patterns of interest. Figure 6.2 shows the result for a timestamped text for which the Kleinberg's algorithm identifies words that burst. In Fig. 6.2, `x` does not burst as it is present in all the years, whereas `y` bursts more than `z`. The algorithm results in a *level of burstiness*, *weight of the burst*, *length of the burst*, and *start and end year* for a

[2] https://ciir.cs.umass.edu/tdt.

[3] https://dl.acm.org/doi/10.5555/944919.944937.

Year	Words
1996	x
1997	x y
1998	x y
1999	x
2000	x
2001	x y y
2002	x y y
2003	x y y
2004	x y y
2005	x y
2006	x y
2007	x y
2008	x
2009	x
2010	x
2011	x
2012	x
2013	x z
2014	x z
2015	x z
2016	x z
2017	x z
2018	x z
2019	x
2020	x
2021	x

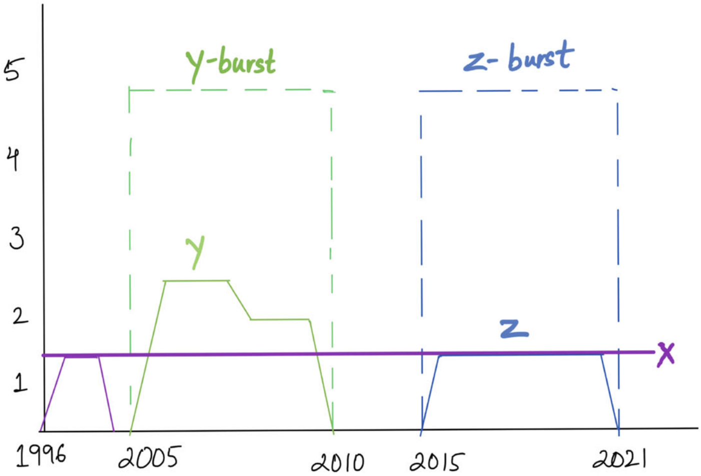

Fig. 6.2 Example showing bursts

given stream of events where every event is a set of keywords with a timestamp. Thus, burst detection identifies time intervals with an unusually high frequency of a specific keyword.

6.1.3 *How to Perform Burst Detection?*

For burst detection, texts need to be cleaned effectively to get a profound representation of the data. The typical steps to perform burst detection are:

1. Pre-processing of text. It includes (i) removal of replies, mentions, URLs, hashtags, and retweets, (ii) correction of spelling errors using a specific dictionary such as Hunspell dictionary, (iii) replacing of abbreviations and shorthand

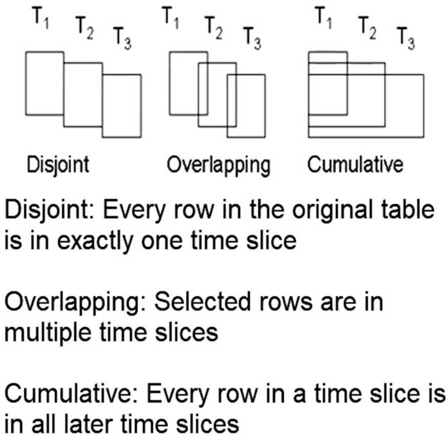

Fig. 6.3 Types of time slicing

notions using a specific dictionary such as SMS (Short Message Service) dictionary for social media data in addition to other pre-processing tasks covered in Sect. 3.3.

2. Filtering of particular time slice by seconds, milliseconds, days, hours, minutes, fortnights, months, weeks, quarters, decades, years, or centuries for disjoint, overlapping, or cumulative time slice (Fig. 6.3); removal of large spikes in the data; normalization of data by de-duplication, unit conversion, and adjusting time zones; integration and interlinking of different data sources; and finally, the aggregation/classification of the data.
3. Performing burst detection.
4. Visualizing the bursts.

6.1.4 Available Tools and Packages

The `Sci2 Tool`[4] and `CiteSpace`[5] are the two open-source tools/applications that non-programmers can utilize to perform burst detection. Programmers can use `Bibliometrix/Biblioshiny`[6] and `bursts`[7] packages in `R`, and

[4] https://github.com/CIShell/sci2/releases/tag/v1.3.0.

[5] http://cluster.cis.drexel.edu/~cchen/citespace/.

[6] https://www.bibliometrix.org/Biblioshiny.html.

[7] https://cran.r-project.org/web/packages/bursts/index.html.

`Metaknowledge`,[8] `burst_detection`,[9] and `pybursts`[10] packages in `Python` to perform burst detection. Most of the tools mentioned above are covered in detail in Chap. 10.

6.1.5 Applications

Burst detection is commonly used to identify activity spikes in email, Twitter, Flickr, Facebook, or news data streams. These bursts of activities can be correlated with external events such as deadlines. It has been applied to several types of documents such as blogs [19, 20], tweets [21–23], spam detection [24], news streams [25, 26], and research articles [8, 27–29]. "Scientific papers tend to enter the world in batches, such as when a new edition of a journal or the proceedings of a conference is published. This violates Kleinberg's underlying assumption that new items enter the dataset in a continuous fashion. It also forces us to impose longer time steps, such as years rather than seconds" [8]. In contrast, several open-access repositories for scientific literature such as PubMed, arXiv, Semantic Scholar, and DBLP contain articles with small intervals compared to the size of the time steps, making them a good data source for burst detection. Bursts can be detected for journal names, author names, keywords, references, terms, or country names used in the abstract or title of a document.

6.1.6 Advantages

1. Scalable.
2. "It can be applied to datasets for which one might have little domain knowledge" [8].
3. It can help "to create a snapshot of the history of a field" [8].
4. It can be helpful "to funding agencies and researchers exploring the research landscape" [8].

6.1.7 Limitations

1. Bursts might pick up trends in language use and style rather than the content of the documents.

[8] https://pypi.org/project/metaknowledge/.

[9] https://pypi.org/project/burst_detection/#:~:text=Burst%20detection%20identifies%20time%20periods,submissions%20to%20an%20annual%20conference).

[10] https://pypi.org/project/pybursts/.

2. Detecting burst events is rather difficult because of noisy and sparse social media texts.
3. Bots or fake accounts spreading misinformation/disinformation by re-posting tweets/news/posts make the information containing them bursty on social media.
4. It might also pick up the changes in the construction of the text, for instance, patents vs. papers.
5. It "requires a span of historical data to detect bursts which means that it cannot effectively detect bursts in the earliest years of a dataset" [8].

6.2 Burst Detection and Libraries

In libraries, burst detection can help to track the trending topics discussed by libraries on Twitter or the subject of popular books being borrowed by library users. Library professionals may use burst detection in bibliometric studies to understand the emergence of a concept or apply it to textual data retrieved from social platforms to determine the burst of a concept/topic as a query over a defined epoch and repackage the information in various formats like a state-of-the-art report to cater to the needs of its users.

6.2.1 Use Cases

6.2.1.1 Personalization

Burst detection can provide tailored content to the patrons matching their preferences and habits based on their profiles, that is, providing them with selective dissemination of information (SDI) service.

6.2.1.2 Information Retrieval

Burst detection can enhance the searching process of large documents in the library to aid in the indexing and ranking process of the library's OPAC.

6.2.1.3 Bibliometrics

Burst detection is a prevalent methodology in the domain of bibliometrics. It has been used to study the fastest rising or hot topics or *bursts* in the scientific literature to provide insight into how they evolved. There are many applications of finding bursty terms in the scientific literature, including (i) "early detection might allow funding agencies and publishers to take note of the most promising new ideas and channel new support that way" [8], (ii) "automatically listing the hottest topics over time would give an instant snapshot of the evolution of the field" [8], and (iii) "compiling a corpus of historical bursty terms over time might make it possible to characterize the life cycles that new ideas go through as they develop" [8].

6.2.1.4 Altmetrics

Burst detection can help the libraries to determine the real-time detection of concepts or keywords appearing in the altmetric data related to the research conducted by their faculty.

6.2.2 Marketing

Burst detection can identify the trending hashtags or topics on the library's social media platforms and help the librarian and decision-making authorities to change/improve library products and services.

6.2.3 Reference Desk Service

Burst detection can identify the most bursty words in a corpus of reference questions asked by patrons in person or virtually and can help the librarian prepare in advance for related queries.

6.3 Case Study: Burst Detection of Documents Using Two Different Tools

6A: Sci2

Problem If you have text documents such as research articles, tweets, newspaper articles, electronic theses and dissertations, blog posts, Facebook posts, or reviews and want to identify the emerging/hot/trending topics in the documents over some time.

Goal To identify the trending/emerging terms, subjects, or topics for a particular period.

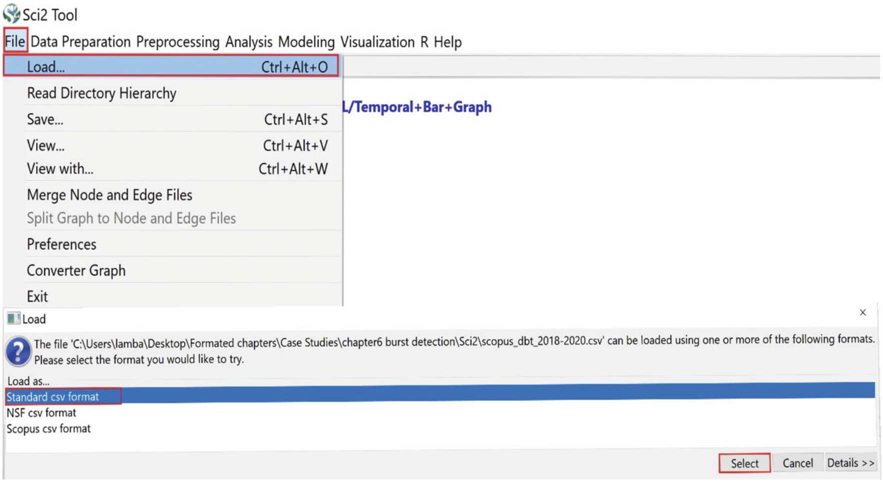

Fig. 6.4 Loading of data in the Sci2 Tool

Dataset A query `"FUND-ALL (department AND of AND biotechnology AND dbt ANDindia) AND (LIMIT-TO (FUND-SPONSOR,"Department of Biotechnology"))AND (LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017) OR LIMIT-TO (PUBYEAR, 2016)) AND (LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO(AFFILCOUNTRY, "India"))"` was searched in the Scopus[11] database for the research articles published with the help of funding received from the *Department of Biotechnology* (an Indian funding agency) in the field of science and technology from 2016 to 2020. A CSV file[12] containing the metadata for 2062 research articles was retrieved from 2018 to 2020.

About the Tool Refer to Chap. 10, Sect. 10.2.7, to know more about `Sci2 Tool`.

Methodology The following screenshots demonstrate the steps which were taken to perform burst detection:

Step 1: The data was loaded into the Sci2 Tool (Fig. 6.4).
Step 2: Text pre-processing was performed on the abstracts (AB) of the retrieved research articles (Fig. 6.5).
Step 3: Burst detection was performed on the pre-processed abstracts (Fig. 6.6).
Step 4: The CSV file consisting of the result of burst detection analysis was saved and edited. In the empty column of "End" year, the last year (i.e., 2020 in this case)

---

[11] https://www.scopus.com/home.uri.

[12] https://github.com/textmining-infopros/chapter6/blob/master/6a_dataset.csv.

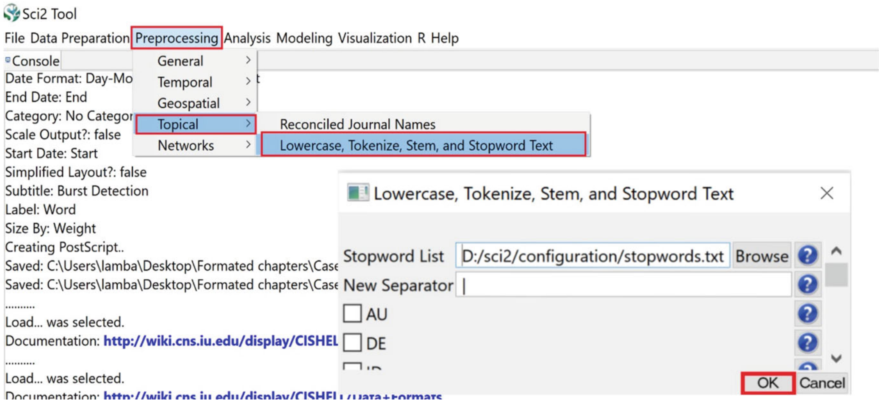

Fig. 6.5 Pre-processing of abstracts in the Sci2 Tool

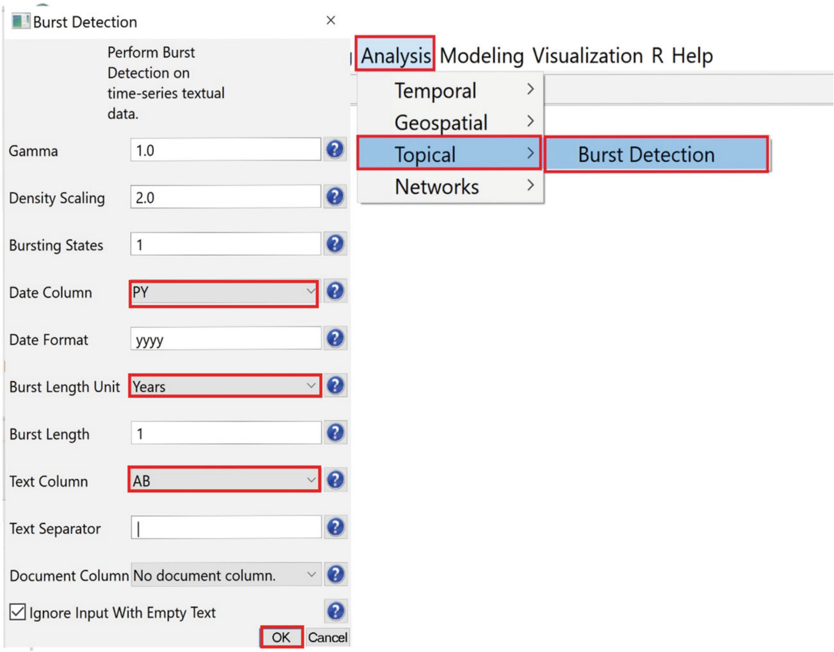

Fig. 6.6 Performed burst detection on abstracts in the Sci2 Tool

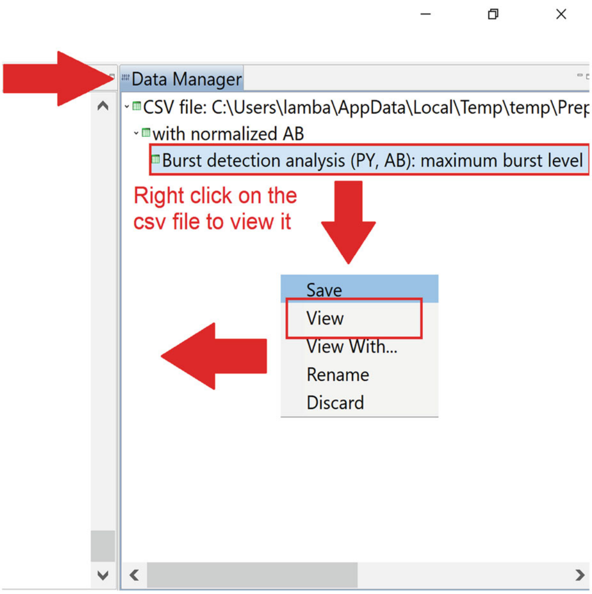

Fig. 6.7 Saving burst detection results

of the analysis was added, and some of the spurious words rows like just, 2019, and 2020 were deleted from the file (Fig. 6.7).
Step 5: The saved CSV file[13] *from Step 4 was then loaded in the Sci2 Tool to visualize the bursts (Fig. 6.8).*
Step 6: The postscript[14] *and bar size*[15] *files were then saved. The postscript file was visualized using psview*[16] *tool (Fig. 6.9).*

Results Figure 6.10 represents each record as horizontal bars with a specific start and end year, where the x-axis is time and the y-axis is amount. The sliced time table[17] shows two levels of bursts: *level 1* shows data from its own time interval, whereas *level 2* shows the growth from one year to another. The top words with the highest weight and area for each year/period and their associated topic/themes are summarized in Table 6.1. As shown in the table, the popular topic of research by Indian researchers in biotechnology in 2018 was on *histones*; in 2019, it was on *breast cancer*; in 2020, it was on *coronavirus*.

[13] https://github.com/textmining-infopros/chapter6/blob/master/6a_maximum_burst_level.csv.
[14] https://github.com/textmining-infopros/chapter6/blob/master/6a_results_horizontal-line-graph.ps.
[15] https://github.com/textmining-infopros/chapter6/blob/master/6a_results_barSizes.csv.
[16] http://psview.sourceforge.net/download.html.
[17] https://github.com/textmining-infopros/chapter6/blob/master/6a_maximum_burst_level.csv.

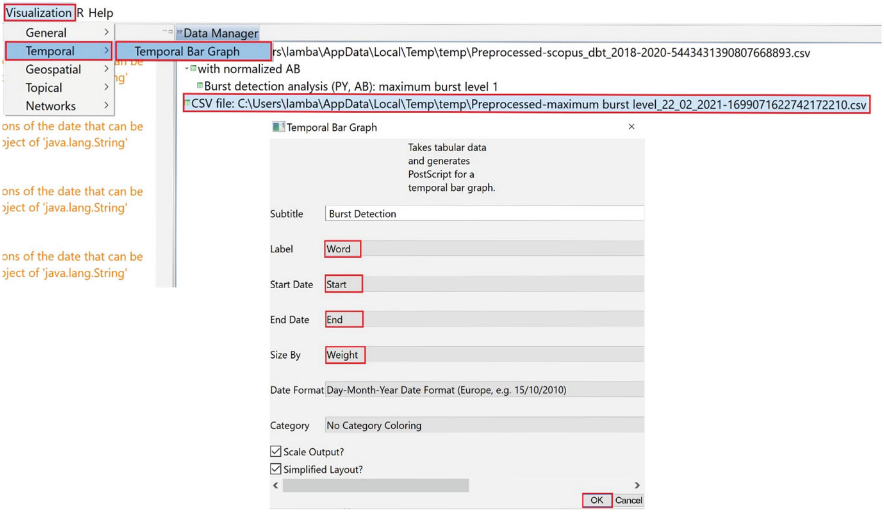

Fig. 6.8 Visualization of bursts

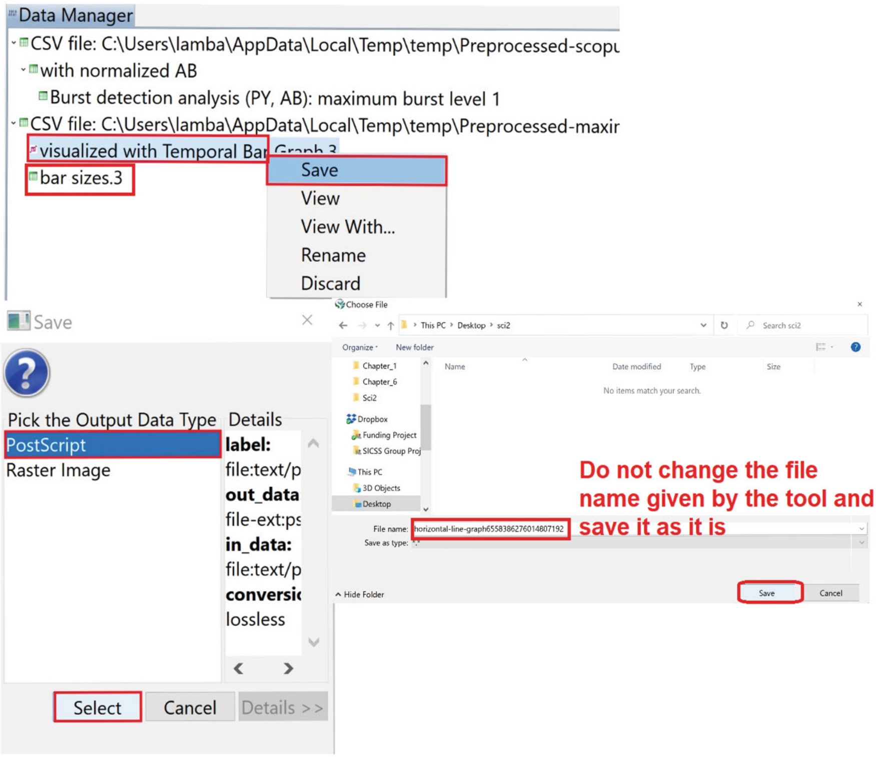

Fig. 6.9 Saving the postscript and bar size files

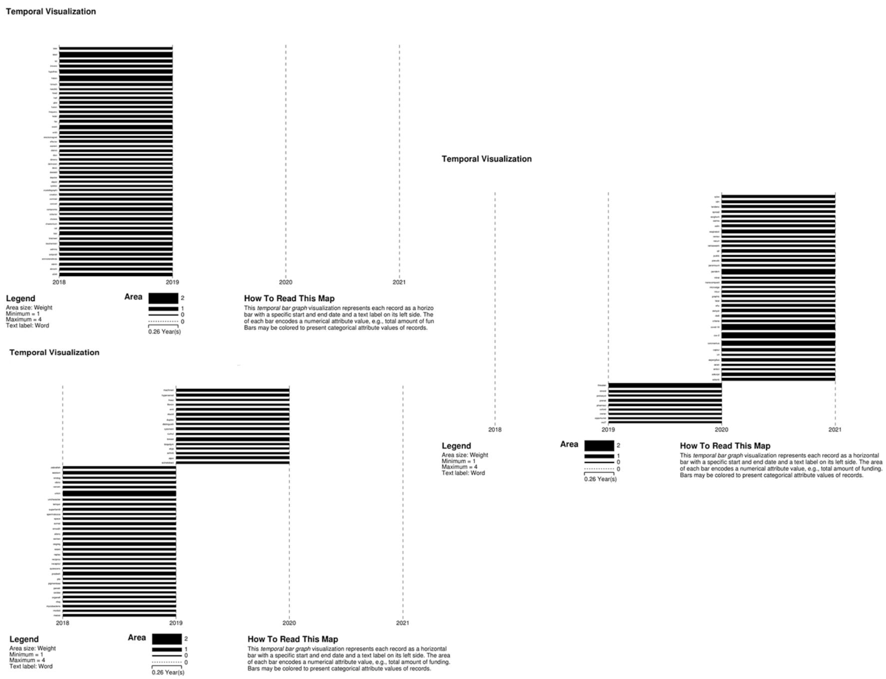

Fig. 6.10 Horizontal line graph

Table 6.1 Summarization of burst results

Level	Period/year	Top 5 words	Area	Topic
1	2018	Label, union, histon, event, hypothes	3.95, 3.93, 3.68, 2.84, 2.78	Histones
2	2019–2020	Threaten, breast, cytometri, machineri, elucid	3.26, 2.60, 2.10, 1.99, 1.95	Breast cancer
1	2020	Covid-19, cov-2, pandem, coronavirus, respiratori	4.23, 4.23, 3.96, 3.43	Coronavirus (COVID-19) pandemic

6B: R

Problem If you have text documents such as research articles, tweets, newspaper articles, electronic theses and dissertations, blog posts, Facebook posts, or reviews and want to identify the emerging/hot/trending topics in the documents over some time.

Goal To identify the trending/emerging terms, subjects, or topics for a particular period.

Prerequisite Familiarity with the R language.

Virtual RStudio Server You can reproduce the analysis in the cloud without having to install any software or downloading the data. The computational environment runs using BinderHub. Use the link (https://mybinder.org/v2/gh/textmining-infopros/chapter6/master?urlpath=rstudio) to open an interactive virtual RStudio environment for hands-on practice. In the virtual environment, open the `burst_detection.R` file to perform burst detection.

Virtual Jupyter Notebook You can reproduce the analysis in the cloud without having to install any software or downloading the data. The computational environment runs using BinderHub. Use the link (https://mybinder.org/v2/gh/textmining-infopros/chapter6/master?filepath=Case_Study_6B.ipynb) to open an interactive virtual Jupyter Notebook for hands-on practice.

Dataset Data was retrieved using an `aRxiv` package that uses arXiv API[18] in R.

About the Tool Refer to Chap. 10, Sect. 10.2.1, to know more about R.

Theory The `bursts` package is "an implementation of Jon Kleinberg's burst detection algorithm [7] which uses infinite Markov model to detect periods of increased activity in a series of discrete events with known times, and provides" [30] visualizations for the results.

Methodology and Results The `aRxiv` library was used to search 1000 abstracts that contained the query "burst detection." 233 articles[19] were identified for the query from 1993 to 2021.

```
#Load libraries
library(aRxiv)
library(bursts)

  data <- arxiv_search('abs:"burst detection"', limit=1000)
write.csv(data, "6b_dataset.csv")
```

`Kleinberg's` function was used to perform burst detection. It resulted in a dataframe that contained a list of all the "bursts" and analyzed the dates when the documents were submitted in the arXiv repository.[20]

[18] https://arxiv.org/help/api/.

[19] https://github.com/textmining-infopros/chapter6/blob/master/6b_dataset.csv.

[20] https://arxiv.org/.

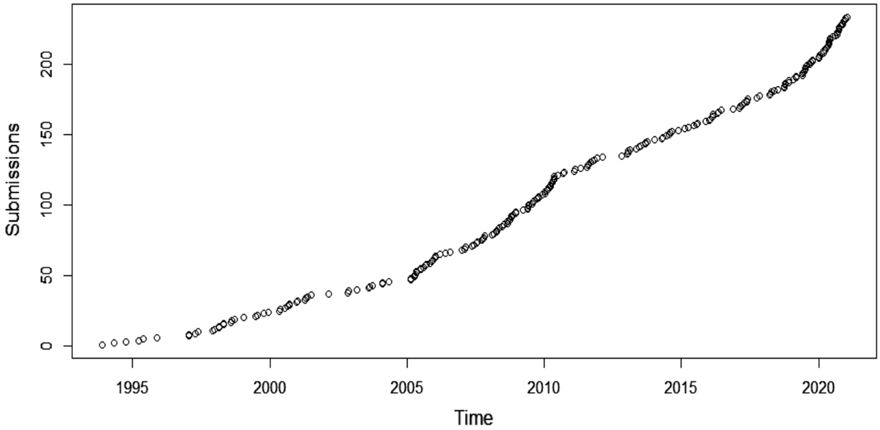

Fig. 6.11 Accumulation of "burst detection" submission in arXiv repository

```
bursts <- kleinberg(as.POSIXct(data$submitted))
```

Figure 6.11 shows the accumulation of "burst detection" submissions in the arXiv repository. It can be observed that submissions in the field of "burst detection" in the arXiv repository are from 1993, and the number of submissions increased in intensity over the years.

```
#Accumulation of submissions
plot(as.POSIXct(data$submitted),
1:length(as.POSIXct(data$submitted)),
xlab='Time',
ylab='Submissions')
```

The bursts were then plotted to present the hierarchical burst structure (Fig. 6.12). Table 6.2 shows (i) the level, (ii) the start date, and (iii) the end date of the identified bursts. Level 1 represents the first event from its first instance (1993) in the arXiv repository to the last (2021). Level 2 represents the additional levels of bursts (2008–2010 and 2018–2021) for the pre-prints in the arXiv repository.

```
#Bursts in submissions
plot(bursts, xaxt = 'n')
axis.POSIXct(1, bursts$start)
```

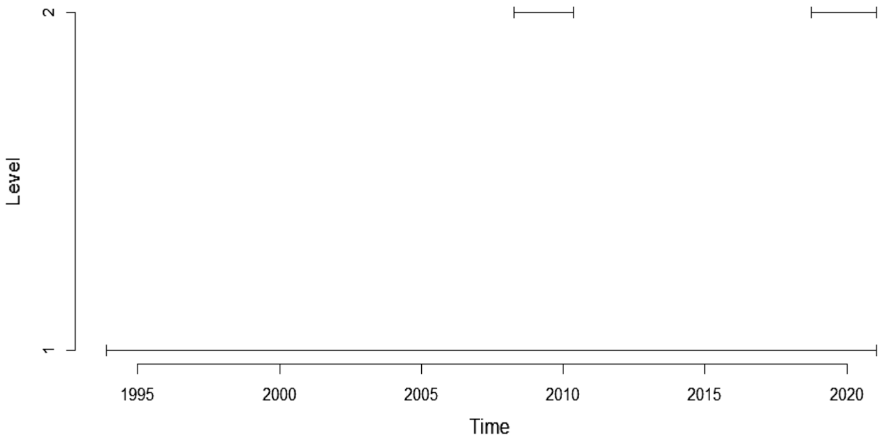

Fig. 6.12 Bursts of "burst detection" in arXiv repository

Table 6.2 Levels of bursts

Level	Start	End
1	11-18-1993	01-12-2021
2	04-07-2008	05-12-2010
2	10-01-2018	01-12-2021

References

1. Lind S (2016) Science of science (Sci2) tool manual. https://wiki.cns.iu.edu/pages/viewpage.action?pageId=1245860#id-4.6TemporalAnalysis(When)-4.6.1BurstDetection. Accessed 22 Feb 2021
2. Zhang X, Shasha D (2006) Better burst detection. In: Proceedings of the 22nd international conference on data engineering. IEEE Computer Society, Washington, DC, pp 146–149
3. Zhu Y, Shasha D (2003) Efficient elastic burst detection in data streams. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 336–345
4. Ryan D (ed) (2004) High performance discovery in time series: techniques and case studies. Springer, New York
5. Singh T, Kumari M (2021) Burst: real-time events burst detection in social text stream. J Supercomput. https://doi.org/10.1007/s11227-021-03717-4
6. Ebina R, Nakamura K, Oyanagi S (2011) A real-time burst detection method. In: 2011 IEEE 23rd international conference on tools with artificial intelligence, pp 1040–1046. https://doi.org/10.1109/ICTAI.2011.177
7. Kleinberg J (2002) Bursty and hierarchical structure in streams. In: 8th ACM SIGKDD international conference on knowledge discovery and data mining. https://www.cs.cornell.edu/home/kleinber/bhs.pdf. Accessed 09 June 2021

8. Tattershall E, Nenadic G, Stevens RD (2020) Detecting bursty terms in computer science research. Scientometrics 122:681–699. https://doi.org/10.1007/s11192-019-03307-5
9. Aggarwal CC, Subbian K (2012) Event detection in social streams. In: Proceeding 2012 SIAM international conference data mining, pp 624–635
10. Carbonell JG, Yang Y, Laferty J, Brown R, Pierce T, Liu X (1999) CMU Approach to TDT-2: segmentation, detection, and tracking. In: Proceedings of the 1999 DARPA broadcast news conference. https://doi.org/10.1184/R1/6604133.v1. Accessed 11 June 2021
11. Lee P, Lakshmanan LV, Milios EE (2014) Incremental cluster evolution tracking from highly dynamic network data. In: 30th International conference on IEEE data engineering (ICDE), pp 3–14
12. Orr W, Tadepalli P, Fern X (2018) Event detection with neural networks: a rigorous empirical evaluation. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Brussels, pp 999–1004
13. McMinn AJ, Jose JM (2015) Real-time entity-based event detection for twitter. In: International conference of the cross-language evaluation forum for European languages, pp 65–77
14. Guille A, Favre C (2015) Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. Soc Netw Anal Min 5(1):18
15. He Q, Chang K, Lim E-P (2007) Using burstiness to improve clustering of topics in news streams. In: ICDM '07: Proceedings of the 2007 seventh IEEE international conference on data mining. IEEE Computer Society, Washington, DC, pp 493–498
16. He Q, Chang K, Lim E-P, Zhang J (2007) Bursty feature representation for clustering text streams. In: Proceedings of the seventh SIAM international conference on data mining, Minneapolis, Minnesota, pp 491–496
17. Lappas T, Arai B, Platakis M, Kotsakos D, Gunopulos D (2009) On burstiness-ware search for document sequences. In: Proceedings of the 15th AC, SIGKDD international conference on knowledge discovery and data mining, New York, pp 477–486
18. Sakkopoulus E, Antoniou D, Adamopoulou P, Tsirakis N, Tsakalidis A (2010) A web personalizing technique using adaptive data structures: the case of bursts in web visits. J Syst Softw 83:2200–2210
19. Kumar R, Novak J, Raghavan P, Tomkins A (2005) On the bursty evolution of blogspace. World Wide Web 8:159–178. https://doi.org/10.1007/s11280-004-4872-4
20. Platakis M, Kotsakos D, Gunopulos D (2008) Discovering hot topics in the blogosphere. In: Proceedings of the 2nd Panhellenic scientific student conference on informatics, related technologies and applications, Samos, pp 122–1332
21. Weng J, Lee B-S (2011) Event detection in twitter. In: Fifth international AAAI conference on weblogs and social media. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2767/3299 Accessed 21 Feb 2021
22. Diao Q, Jiang J, Zhu F, Lim EP (2012) Finding bursty topics from microblogs. In: Proceedings of the 50th annual meeting of the association for computational linguistics: long papers-volume 1, ACL '12, pp 536–544
23. Mathioudakis M, Koudas N (2010) Twittermonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD international conference on management of data, SIGMOD '10, pp 1155–1158
24. Xie S, Wang G, Lin S, Yu PS (2012) Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12. ACM Press, Beijing, p 823
25. Fung GPC, Yu JX, Yu PS, Lu, H (2005) Parameter free bursty events detection in text streams. In: Proceedings of the 31st international conference on very large data bases, VLDB '05, pp 181–192
26. Takahashi Y, Utsuro T, Yoshioka M, Kando N, Fukuhara T, Nakagawa H, Kiyota Y (2012) Applying a burst model to detect bursty topics in a topic model. In: Isahara H, Kanzaki K (eds) Advances in natural language processing, Berlin, pp 239–249

27. Pollack J, Adler D (2015) Emergent trends and passing fads in project management research: a scientometric analysis of changes in the field. Int J Proj Manag 33:236–248. https://doi.org/10.1016/j.ijproman.2014.04.011
28. He D, Parker DS (2010) Topic dynamics: an alternative model of bursts in streams of topics. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, pp 443–452
29. Mane KK, Börner K (2004) Mapping topics and topic bursts in PNAS. Proc Natl Acad Sci USA 101:5287–5290. https://doi.org/10.1073/pnas.0307626100
30. Binder J (2015) Bursts. https://cran.r-project.org/web/packages/bursts/bursts.pdf. Accessed 13 June 2021