Information Labelling of Medical Forum Posts by Non-Clinical Text Information Retrieval

Amit Kumar Kushwaha, Arpan Kumar Kar

Indian Institute of Technology Delhi, New Delhi, India

Abstract

With the advent of web 2.0, modern societies produce a vast amount of data, and merely keeping up with storage and transmission is difficult; analyzing it to extract useful information has become further challenging. All the historical research in healthcare data processing is more concentrated on formal clinical data. There lies a lot of valuable yet idle lying data in the non-clinical information as well. The proposed study combines the state of the art methods within distributed computing, text retrieval, clustering methods, and finally, using a classification method to a computationally efficient system that can clarify cancer patient trajectories based on non-clinical and freely available online forum posts. The motivation is that informed patients, caretakers, and relatives often lead to better overall treatment outcomes due to enhanced possibilities of proper disease management. The resulting software prototype is fully functional and built to serve as a test bench for various text information retrieval and visualization methods. Via the prototype, we demonstrate a computationally efficient clustering of posts into cancer-types and subsequent within-cluster classification into trajectory related classes. The system also provides an interactive graphical user interface allowing end-users to mine and oversee the valuable information.

Keywords 1

Machine Learning, Chatbot, Artificial Intelligence, Medical, Ontology

1. Introduction

Most of the patients that acquire a progressive and terminal disease are towards the latter end of life. Some of these illnesses are primarily respiratory disorders, cancers, and cardiovascular. This illness implies a large time frame for the patients themselves and the surrounding relatives and caretakers [1], [2]. The trajectory of the timeframe can be summarized as a sequence of steps shown in figure 1. Although the entire trajectory looks very simple and compact, they are complex and contain a range of concerns underneath each of the four steps. For instance: life expectancy at each stage, patterns of decline, probable interactions with other health services, medicinal side effects, treatment plans and costs at each stage, any other non-documented side effects, palliative care, and many more.

Mis-informed outputs can lead to costlier yet un-successful and delayed treatment. Scholarly outputs, in turn, can have clarified trajectories of the timeframe and can lead to better overall treatment owing to better clinical sources and decisions. This further reduces the possibilities of fewer re-admissions, decreased health care costs, and higher quality of life for patients in the potentially final weeks, months, and years. Unlimitedly, better overall care is obtainable via clarification during early stages, estimation, and communication of patient-specific symptoms and disease trajectories.

The proposed study is motivated by the idea of exploiting the relevant yet idle information in the ever-increasing user-generated content

ISIC'21: International Semantic Intelligence Conference, February 25-27, 2021, Delhi, India
EMAIL:Kushwaha.amitkumar@gmail.com; Arpan_kar@yahoo.co.in
ORCID: 0000-0002-5537-1250; 0000-0003-4186-4887

through online and freely accessible non-clinical text for the benefit of anyone interested in any clinical trajectory, e.g., cancer patients, COVID-19 patients. With the recent increase and rise in the overall COVID-19, there was massive unrest during the initial stages of the disease spread, where even the patients who were tested positive were not sure about the trajectories of the sequence of steps in figure 1. This motivated me to further this study, which can be an essential literature contribution for researchers and act as a day-to-day practice implication for someone who has internet but cannot navigate through much-unstructured to find simple, relevant clinical information.

Historical data shows that approximately one-third of the entire world's population gets diagnosed with cancer during their lifetime [3]. According to the World Health Organization (WHO) [4], as of 9th August 20 globally, there have 20 million patients tested positive for COVID-19. Thus, a large community of potential end-users can consume the non-clinical data for answering their queries related to clinical trajectories. A cancer diagnosis or in recent times COVID-19 leads to several reactions, a predominantly one is first sought information online on specific symptom, type and severity and finally trajectory prognosis.

A trend that has recently gained prominence among the community is to communicate on online forums [5], [6], [7], [8], [9], [10]. On these medical forums, people have the right to write freely on their emotions and what they feel about the disease, treatment, after-effects, and normalcy after the treatment without disclosing the identity. For instance: on cancer forums, people write freely about their initial stage frustrations, fears, and how they overcame them. The same applies to COVID-19 forums too. Any healthcare system does not leverage this freely available non-clinical, nonetheless less potentially very relevant information.

Mining all such relevant information from a wide variety, volume, and veracity of online user-generated content on the forums is an overlap of the technical-scientific research domain. It is more challenging than mining standard health texts such as electronic health records (EHR), including hospital admission journals that capture doctors' comments, medical reports, and similarly discharge summaries. In all these formal EHRs, the language of cause, symptoms, cures, and after-effects is more concise, specific, and medical terms are used more distinctly from case to case. These terms are way different from a layperson's mention of terms in the same context on the online forums. This adds to our motivation to make this non-clinical data available for a person in general.

The current research objective is to clarify and communicate the patient trajectories at each stage by computationally efficient text information retrieval from non-clinical online forum post texts. Through the current study, the identified research objectives are met by building a fully functional and generalizable framework that can screen/filter, process, and present the non-clinical data for clinical trajectory in a visually and informative way. The framework is chosen to act as a test bench for future text information retrieval methods and is not only restricted to the current study. The current study's underlying premise is unstructured inherent and valuable information, which is freely available on non-clinical yet medical forums.

2. Related work

Scholars' research with the objectives, methods, and hypothesis rooted in data mining has been mostly focusing on text summarization. In 2005, Murray et al. [2] performed the clinical review research that summarizes three disease trajectories: organ failure (heart and long), frail elderly, and cancer. In another related study in 2010, Ebadollahi et al. [11] predicted a patient's trajectory from temporal physiological data. This study was further improved in a 2014 research undertaken by Jensen et al. [12] with the disease trajectory data spanning fifteen years from a large patient population.

In 2016, Ji et al.[13] proposed a predictive model for health condition trajectory and co-morbidity relationships by training the social health records model. Another related study was performed in 2017 by Jensen et al. [1] using text analysis using EHRs to predict patient (cancer) trajectories automatically. However, summarizing all the above work, we interpret a gap in text information retrieval using distributed clustering and classification. None of the highlighted studies have ventured using this framework, which can be computationally efficient. At the end of the proposed current framework, a classification model can quickly identify patient trajectories using non-clinical texts from online forums.

Frunza et al. [14] did a related study in 2011; in their study, they automatically extract sentences from clinical papers about diseases and treatments. Based on the extracted sentences, semantic relations between diseases and associated treatments are then identified. Another related study was done by Rosario et al. [15] in 2004. The focus of their work was to recognize text-entities containing information about diseases and treatments. They use Hidden Markov Models and Maximum Entropy Models to perform the entity and disease-treatment relationship recognition.

Compared to Frunza et al., the later work focuses mostly on classification. In the proposed study, the present study also focuses on text retrieval and clustering through the current study. The current proposed study will also focus on cancer and COVID-19 trajectories, where-in the other studies have only focused on cancer as a prevalent disease.

Lastly, in the 2011 study by Yang et al. [16], Density-Based Clustering was used to identify topics within online forum threads on social media. They also developed a visualization tool to provide an overview of the identified topics. Their tool's purpose was to extract topics with sensitive information related to terrorism or other criminal activities; however, it might also be tailored to extract other topics. Besides using DBSCAN, the study proposed a related clustering method, namely SDC (Scalable Density-based clustering). The structure of the Yang et al. study is, to some extent, as the present study; individually, in the present study, topics are also extracted from online forum posts, density-based clustering is also used, and result visualization capabilities are also provided.

2.1. Research gap addressed by the novelties of the current work

The novelty of the proposed study is combining the state of the art un-supervised distributed computing text retrieval through clustering. This contribution is topped by a coherent classification that is computationally efficient and can identify patient trajectories based on non-clinical texts. Hence, the outcome of the proposed study provides a unique and novel means for an individual and researchers looking for cancer and COVID-19 trajectories. This is done by activating relevant and potentially hitherto overlooked, by the established health care systems, information hidden in non-clinical texts.

2.2. Significance

In general, computational retrieval of information from the vast amounts of health care texts is significant. Specifically, for this study, the significance lies in the systematic combination of state-of-the-art methods to mine, refine, categorize, and present laypersons' cancer trajectory related descriptions. It is significant to empower patients and caretakers and help build healthy patient/caretaker communities by leveraging the soft information not hitherto used by the established health care systems, e.g., information about emotions, feelings, or personal preferences.

3. Proposed framework

3.1. Overview

The proposed study has four major building blocks or components, including a database component for storing the cluster outputs. A detailed visual representation of the framework is given in figure 1 below. It has been designed in a micro-service architecture with one process per component to make the framework light from a production implementation standpoint.

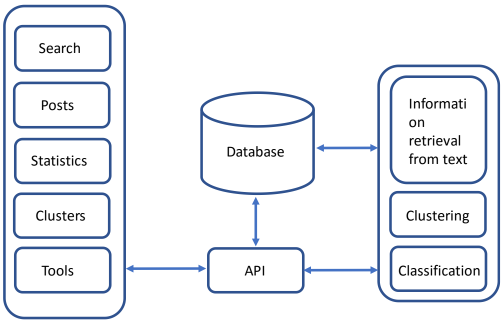
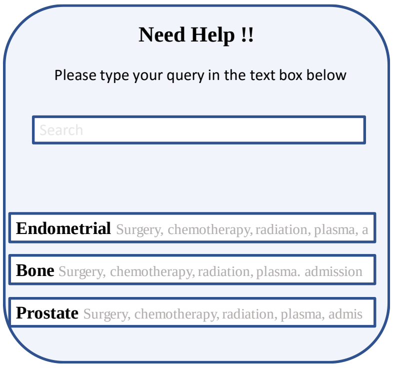


Figure 1: Framework

The left part of the framework in figure 1 above is the front-ending component that handles the user interaction. We will be further elaborating the same in section 3.2. The API component's sole purpose is to enable the front end component to interact with the database and with other service components. The Database component persists all gathered forum posts and the computed results, e.g., clusters, classes, and cancer-trajectories. The Service component handles the computationally burdensome data processing; the micro-service architecture enables scaling of this component only. Implementing the service component as a scalable unit becomes well-suited for the application of a distributed computing approach. Especially the clustering calculations are burdensome and need to be made efficient. Currently, the text retrieval and classification calculations do not need to be scaled as they are much faster than the clustering.

3.2. Front-end

Having a front end to interact with data helps to explore results from the end-user's perspective. The developed user interface is useful for exploring the collected data set of forum posts and to show information from an area of interest. For instance: a user can select a cluster, i.e., a disease-type, of interest, e.g., COVID-19 or lymphoma cancer, and only receive posts within that cluster. A user can also choose a pivot of information, e.g., side effects of COVID-19 medicine or side effects of cancer radiation, and thereby see all posts from the cancer cluster or COVID cluster that contains information about side effects. Such a tool is relevant for scientific use and cancer patients and caretakers.



Need Help !!

Please type your query in the text box below

Search

Endometrial Surgery, chemotherapy, radiation, plasma, a

Bone Surgery, chemotherapy, radiation, plasma. admission

Prostate Surgery, chemotherapy, radiation, plasma, admis

Figure 2: Front end search view

The user interface consists of five main views: Search, Posts, Statistics, Clusters, and Tools (figure 2). In the Search view, a user can search the entire collection of forum posts, the identified clusters. Initially, a view of the types of clusters, as shown in figure 3, will be displayed to the end-user. By clicking a type cluster, all posts associated with that type of cluster is displayed in the Posts view. Users can browse through the posts within a type of cluster and by selecting a class-label.

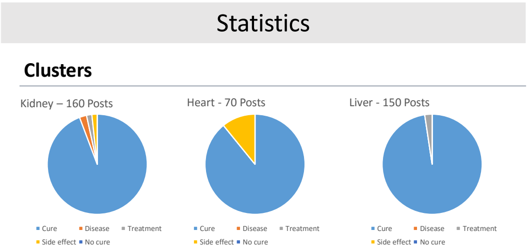


Figure 3: Cluster view

3.3. Functional validation of the outputs

The robustness of a framework is considered based on the statistical metrics and needs to be measured on the intuitiveness of the text outputs as received by the users. Hence in order to concretely measure the outputs of the framework, we check the output from the below qualitative lens as well other than the statistical metrics:

Functional intuitiveness:

- Appropriate,
- Suitable

Performance:

- Time,
- Utilization

Compatibility:

- Coherence

Scalability:

- Modular structure,
- Easily modifiable

Portability:

- Easy installation

4. Information retrieval

4.1. Data collection

The data-set has been created by collecting the texts from online posts on the medical forums non-clinical. The posts are mostly written by person in-general and not doctors or medical staff. Hence the topics and words used are more day-to-day life and less skewed towards specific medical terminologies. Typically, the data collected from posts will consist of symptoms, initial experiences, treatments, place where treated, post-treatment experience, questions, side-effects, and outcomes. The most informative and unstructured data is stored in the actual text of each row. This text's basis, the information retrieval framework proposed in the current study, extracts the relevant features for clustering. Often, these non-clinical texts captured contain rather detailed descriptions of a disease (like cancer or COVID-19) and the specific treatment received.

4.2. Data preprocessing

To ensure that the actual text information retrieval works successfully, the collected text needs to be preprocessed and cleansed for any noise in the data. For the proposed research, we have conducted three preprocessing steps:

1. Cleansing,
2. Stemming, and
3. Tokenization

The first step of cleansing consists of processes to remove unwanted characters, e.g., HTML tags, emojis, and ASCII-artworks. This is a non-trivial task when dealing with forum posts as people express themselves quite informally. In the second step of the stemming part, inflected and derived words are reduced to their word stem [17]. Different algorithms for stemming exist in the literature, e.g., the Lovins Stemmer [18], the Paice Stemmer [19], and the predominant Porter Stemmer [20]. All these stemming algorithms are best suited for English; in the present study, the Porter Stemmer is used. The Porter Stemming algorithm is based on five steps, and in each step, a specified set of rules is applied to the word being processed. For instance, the first step contains the following processing rules, as represented in figure 4. In the tokenization part, character and word sequences are sliced into tokens. Typically, the tokens are words or terms, but in this study, tokens are only words. After the tokenization, stop words are removed.

4.3. Information retrieval

To make sure that the clustering of posts into a specific type of disease clusters to be accurate, information from all the collected posts' content attributes must be extracted. This is achieved by using various natural language processing [9] and text retrieval, together with a predefined feature vector containing names of a range of disease types. For current work, we use the term weighting approach. This approach uses term frequency and inverse document frequency to yield term frequency-inverse document frequency, which is the term's final weight. The purpose of term frequency (tf) is to measure how often a term occurs in a specific text corpus, i.e., in this study, tf is simply an unadjusted count of term appearances.

Term frequency [21] can be defined as tf(t,d) | occurrences of term t in document d. Documents vary in length, which entails a bias in tf; that is, a term is likely to appear more often in a lengthy document than in a short document, given the documents are similar in content [22]. Whenever a term is frequent in a document, it is likely to be relevant to that specific document. The purpose of inverse document frequency (IDF) is to measure the weight of a term in a collection of documents; a rare term is often more valuable than a common term in a collection of documents [23].

Term frequency-inverse document frequency (*tf-IDF*) is a measure of how important a word is to a specific document in a collection of documents. A significant tf-idf weight is obtained whenever: 1. the term frequency is high for the specific document, and 2. the document frequency is low for the term across the collection of documents. Combining the tf and IDF weights tends to filter out standard terms that do not carry much information [24], [25].

5. Clustering

5.1. Existing DBSCAN clustering

Clustering is a process of grouping unlabeled data into clusters of homogenous attributes. The data points in each cluster have similar traits, such that the variance within-cluster is minimum, and variance across clusters is maximum. In the proposed study, a cluster would represent a homogenous group of similar texts from posts. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm based on data points' density (also known as observations). DBSCAN helps to create clusters with a high density of data points, and in doing so, it allows clusters of any shape even if it contains noise, which is slightly different in approach compared to conventional clustering algorithms.

DBSCAN can now find clusters of different sizes and skip the input of taking the number of clusters beforehand. In DBSCAN, the *$\mathcal{E}$-neighborhood* of point *p* will be defined by the points within a radius $\mathcal{E}$ of *p*. If a point *p's $\mathcal{E}$-neighborhood* contains at least m_{pts} number of points, the point *p* is called a core point. A data point is called noise if it is not a core point. A point *p* is in the density-range from a point *q* if *p* is within the *$\mathcal{E}$-neighborhood* of *q*, and *q* is a core point.

A point *p* is defined as in the density range from a point *q* with regard to $\mathcal{E}$ and $\mathrm{m}_{\mathrm{pts}}$ if there is a chain of points, p_1,…..,p_n, where $p_1 = q$ and $p_n = p$ such that *pi+1* is in direct density range from *pi*. A point *p* is defined as a density-connected point to another point *q* with regards to $\mathcal{E}$ and $\mathrm{m}_{\mathrm{pts}}$ if only there is a point *o* such that both *p* and *q* are density-reachable from *o*. A point *p* is a border point if *p*'s $\mathcal{E}$ neighborhood contains less than $\mathrm{m}_{\mathrm{pts},}$ and *p* is in direct density from a core point. A cluster *C* is a non-empty set that satisfies the following two conditions for all point pairs (*p;q*):

1. If *p* is in *C* and *q* is density-reachable from *p*, then *q* is also in *C*; and
2. If (*p; q*) is in *C*, then *p* is density-connected to *q*.

To create a cluster, the DBSCAN algorithm initiated an arbitrary point *p* and searched for all the points in the density range of *p* with respect to $\mathcal{E}$ and $\mathrm{m}_{\mathrm{pts}}$. If *p* is a core point, then a new cluster with *p* as a core point is created. If *p* is a border point, DBSCAN browses the next point in the sample. DBSCAN can also merge any two clusters into one of these clusters are in the same density range. The algorithm will converge when no new points can be added to any existing or new clusters.

5.2. MapReduce DBSCAN clustering

The entire process of DBSCAN clustering is computationally costly with high time and memory consumption. To reduce this consumption and increase efficiency, MapReduce DBSCAN was proposed. The only difference between a regular DBSCAN clustering and DBSCAN via MapReduce is through distribution computation. The steps followed in a MapReduce DBSCAN can be shown in figure 4 below.

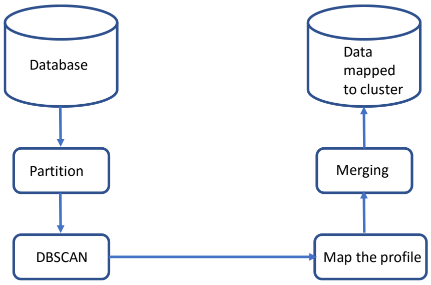


Figure 4: MapReduce DBSCAN

5.3. Partition in MapReduce DBSCAN clustering

To maximize runtime efficiency through invoking, parallel processing can be achieved if the data is well balanced. If the data is well balanced, then the computational load can be evenly distributed on computer nodes' execution. In real-life text data, it is usually un-balanced, and the best strategy to deal with this is using data portioning. This is an inherent part of MapReduce DBSCAN.

Recursive split is the best and frequently used data partitioning method, which helps split the entire bigger data-set into smaller subsets. This is done recursively till a stop criterion is met. All partitions then contain less than a given number of points, or a given number of partitions have been made. Logically, a partition cannot be smaller than 2Ɛ; when a partition is split, the geometry must remain extended beyond 2$\mathcal{E}$. When splitting a partition into two in MapReduce DBSCAN, all possible splits are considered. The split that minimizes the loss in one of the sub-partitions is chosen. Here, the loss is calculated as the difference between the number of points in sub-partition-1 and half of the number of points in sub-partition-2. Each partition is given a key and associated with a reducer.

5.4. Local DBSCAN

Continuing the definition of reducer from the previous paragraph, each reducer will be given a partition and all its associated data points, and hence a mapper should prepare all data related to a partition. Explaining the same concept using an example: the data assigned would be the related data Ci within Pi, and the data within Pi's Ɛ-width extended partition Ri that overlap the bordering partitions.

Local DBSCAN borrows the working principles from the original DBSCAN to perform the clustering. It starts with an arbitrary data point *p* belonging to *Ci* and searches for points in the density of *p* with respect to $\mathcal{E}$ and m_{pts}. If *p* is a core point, the $\mathcal{E}$ neighborhood will be explored for data points. If Local DBSCAN finds a point in the outer margin directly in the density range from a point in the inner margin, it is added to the merge-candidate set. If a core point is in the inner margin, it is also added to the merge-candidate set. Each point in the cluster is given a local cluster-id generated and mapped from partition id and the label id from the local clustering.

5.5. Mapping profile

After each partition has undergone clustering and merge candidate lists have been generated, the merge candidate lists are collected to a single merge candidate list. The basics of merging the clusters from the different partitions are 1. Execute a nested loop on all points in the collected merge candidate lists to see if the same data points exist with different local cluster IDs; 2. If found, then merge the clusters.

Figure 5 illustrates two examples of cluster-merge propositions. Example 1: the points d_1 belong to C_1, and d_2 belong to C_2 are core points, and d_2 is directly density-reachable from d_1; thus, C_1 should merge with C_2. Example 2: The point d_3 belongs to C_1 is a core point, and *r* belongs to C_2 is a border point; thus, C_1 should not merge with C_2.

Mapping Profile step where the purpose is to create a profile that maps clusters that should be merged. The algorithm for generating the mapping profile is represented in the algorithm in figure 5. The output of the algorithm is a list of pairs of local clusters to be merged (denoted MP) and a list of border points (denoted BP); a point *p* is at least a border point in a merged cluster (this is taken care of in the next step).

```
for each cp in CP do
    for each bp in BP do
        if $cp.id == bp.id$ then
            MP.add (($cp.local\ cluster\ id$),
                    ($bp.local\ cluster\ id$))
            BP.delete(bp)
        end if
    end for
end for
```

Figure 5: Merge mapping

5.6. Merge

The previous step resulted in a list of pairs of clusters to be merged. The IDs of the local clusters should be changed into a unique global ID after merging. Thus, a global perspective of all local clusters is built (algorithm in figure 6). Lastly, as mentioned in the previous step, noise points are set to border points.

```
for each element pair $e_i , e_j$ ε MP; i≠j; do
    if $e_i , e_j$ $\mathcal{E}$ L then
        put $e_i$ and $e_j$ into the same Map Slot in L
    end if
    if $e_i$ ε L ∧ $e_j$ ε L then
        put $e_j$ into $e_i$'s Map Slot in L
    end if
    if $e_i , e_j$ $\mathcal{E}$ L then
        if $e_i$ and $e_j$ are not in the same Map Slot in L,
        then move the Map Slot with the highest index to
        the Map Slot with the lowest index
    end if
end for
return L
```

Figure 6: Global ID map

6. Classification

The result of the clustering is a set of specific disease type clusters. To enable further filtering possibilities for the end-user, a within-cluster classification is conducted such that each post within a disease type cluster is labeled with one of the six labels illustrated in table 1. This allows an end-user to filter the forum posts such that, for instance, only posts with specific disease (cluster) treatments (class) are shown.

We have chosen to classify with a Naive Bayes classifier trained with a manually created training set augmented with the freely available set from the BioText Project, UC, Berkeley [26]. The Frunza et al. study also uses a Naive Bayes classifier with promising results [9]. However, they classified abstracts from scientific articles, which is a somewhat different data-domain than the present study's non-clinical texts. The time complexity for training a Naive Bayes classifier is O(np), where n is the number of training observations, and p is the number of features; thus, disregarding the constant, the complexity is in terms of observations O(n). When testing, Naive Bayes is also linear, which is optimal for a classifier.

Table 1
Results

Class label	Class description with example posts in italics
Cure	About cancer-curing treatments. *After 16 chemo sessions, my cancer was gone*.
No cure	About cancer non-curing treatments. *My husband went through chemo since he had bladder cancer. Sadly, he passed*.

6.1. Clustering

MapReduce DBSCAN is a distributed extension of DBSCAN, and they use the same principle for clustering. Thus, given the same input, the two clustering methods should yield the same output. The results in this section show that this is indeed the case, and we thereby consider the implementations of MR-DBSCAN and DBSCAN to be verified in terms of the correctness of the logical output. The actual implementations do not share code, so it seems fair to disregard the odd risk of having both implementations wrong in a manner that lead to the same output.

For comparing the clustering results of DBSCAN and MR-DBSCAN, the Adjusted Rand Index (ARI) [30] is used. The index is a similarity measure between two clusterings, and it is obtained by counting the number of identical labels assigned to the same clusters vs. the number of identical labels assigned to different clusters. If the label assignments coincide fully, the index is 1, and if they do not coincide at all, the index is 0. If DBSCAN and MR-DBSCAN are implemented correctly, the ARI must be one regardless of: 1. the number of points in the data set, 2. The number of partitions in MR-DBSCAN, and 3. the parameter settings for Ɛ and m_{pts}. Also, the number of partitions (#P) in MapReduce DBSCAN, the coverage percentage (%C), and the number of labels (#L) in DBSCAN and MapReduce DBSCAN have been recorded. The results show (Table 2) that the ARI is 1 in all 18 test cases; a necessary condition for this is that both MR-DSBCAN and DBSCAN yield the same number of labels in all the tests also the case (table 2).

Also, MR-DBSCAN has been partitioning its data into 3-8 partitions (table 2), which means that even though the data has been split and clustered individually per partition, the merging works as intended and yields the same clustering as DBSCAN. The coverage percentage value is also identical for the two clusterings in all test cases.

Table 2
Adjusted rank index of clustering

Posts	e	M_{pts}	DBSCAN		MapReduce DBSCAN			ARI
25000	10-3	5	10	3.34	10	3.34	8	1
25000	10-3	50	2	2.99	2	2.99	8	1
25000	10-3	100	1	2.66	1	2.66	8	1
25000	10-2	5	10	3.34	10	3.34	7	1
25000	10-2	50	2	2.99	2	2.99	7	1
25000	10-2	100	1	2.66	1	2.66	7	1
25000	10-1	5	11	3.37	11	3.37	3	1
25000	10-1	50	2	2.99	2	2.99	3	1
25000	10-1	100	1	2.66	1	2.66	3	1
35000	10-3	5	23	2.92	23	2.92	7	1

35000	10-3	50	2	2.37	2	2.37	7	1
35000	10-3	100	1	2.02	1	2.02	7	1
35000	10-2	5	23	2.92	23	2.92	6	1
35000	10-2	50	2	2.37	2	2.37	6	1
35000	10-1	100	1	2.02	1	2.02	3	1

6.2. Real-time analysis of MapReduce DBSCAN

The motivation behind the proposed study is to demonstrate the real-time application of each of the MapReduce DBSCAN steps under variations in

1. the number of forum posts, and
2. the neighborhood radius Ɛ.

These two parameters have the most significant influence on MapReduce DBSCAN's runtime. The Ɛ parameter is used when partitioning the data set, and therefore, it directly influences the beneficial effects of MapReduce. In all tests, the lower point-count threshold for establishing a core point, m_{pts}, is fixed to 5 points. This is done as the parameter only has very little runtime influence, and this influence is isolated to the DBSCAN step, i.e., it does not highlight runtime differences between DBSCAN and MapReduce DBSCAN.

For all 30 test cases (table 3), mapping takes almost no time; merging has also only a little effect on runtime. For relatively large values of Ɛ, i.e., 1 and 0.1, compared to the data span, MapReduce DBSCAN cannot partition the data set well. This affects the runtime as the clustering is then performed on a single partition (or very few), and no MapReduce improvements are achieved. For relatively small values of Ɛ, i.e., 0.001 and 0.0005, the data set is split well into partitions, but due to the low value of Ɛ there are many possible partitions, and much time is spent in search of the best partitioning. Thus, as the results show, the partitioning becomes slower when " decreases, but the local DBSCAN becomes faster. Hence, Ɛ needs to be set with care to strike a balance and minimize the total runtime of MapReduce DBSCAN. In our experiments, the balance is Ɛ = 0.01; here, the partitioning runtime is relatively low, and likewise for the local DBSCAN; this results in a relatively low total runtime.

6.3. Validation of clustering

The purpose of this experiment is to compare time as a function of the number of forum posts of the three different clustering algorithms DBSCAN, MapReduce DBSCAN, and Hierarchical Density Estimates DBSCAN. Algorithm parameters are fixed and equal across the tests in order not to bias the results. Specifically, the lower point-count threshold for establishing a core point mpts = 50 and the neighborhood radius Ɛ = 0.01 for all tests. Note that the setting Ɛ = 0.01 was previously found (section 7.2) to be a suitable choice for MapReduce DBSCAN. The data set in this experiment are various subsets of the collected forum posts; the number of *tf-idf* features has been limited to 1000. The results of all tests are reported in table 3 and figure 7.

Table 3
Results of various clustering

Posts	MapReduce DBSCAN [s]	DBSCAN [s]	Hierarchical Density Estimates DBSCAN [s]
10000	11.696	4.755	11.969
20000	19.545	21.167	48.731
30000	31.115	50.237	105.007
40000	37.321	92.392	217.033

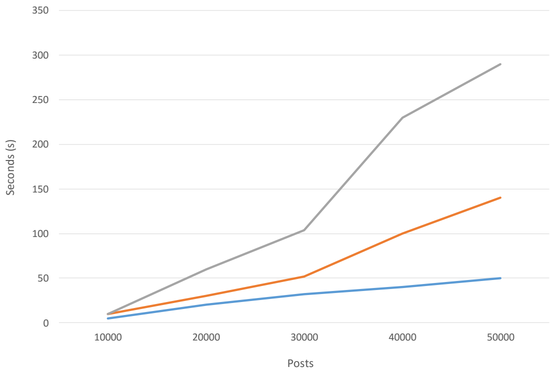


Figure 7: Comparison of clustering DBSCAN (blue), MapReduce DBSCAN (red), and Hierarchical Density Estimates DBSCAN (green)

7. Discussion

The primary motivation of the proposed work and research undertaken to mine the clinical or medical information from non-clinal posts collected from forums is valuable and worth making available to others in a more structured form. In the proposed study, this is achieved by a decision support system that can act as a source of information to help any disease patients like COVID-19, cancer and their caretakers and families to learn about the disease trajectories, initial symptoms, diagnoses outcomes, sources, treatment centers, treatment is taken, after-effects of treatment and costs.

Through the non-clinical posts on forums, the information retrieval framework using text-retrieval, unsupervised clustering, and a classification model. The framework is designed to execute on a distributed computing set-up like MapReduce to increase computational efficiency. The response time of a computationally costly clustering on texts improves a lot, needed for a real-time application.

Moreover, the endpoint of the current framework to the customer is a user interface that enables the end-user to interact with the database and mine for valuable information to understand the overall trajectory of any disease. This helps the patient be in a frame of mind before getting a doctor's consultation and word. This framework will also mobilize online social communities of patients and their caretakers, families using soft information and non-clinical, hitherto conversations.

The proposed framework through the study is an excellent contribution to the existing literature in several different ways. Adding, refining, and benchmarking more clustering and classification methods would yield more comprehensive information through non-clinical texts that might lead to better results, i.e., more accurate clustering and classifications, and thus, ultimately, a better end-user service. The classification would mainly be of interest to collect and use a more extensive training set. The response time of DBSCAN and Hierarchical Density Estimates DBSCAN clustering has been improved by redesigning the algorithms to guarantee upper bounds on memory consumption. This can act as a reference in literature for future researchers.

Lastly, in conclusion, the proposed system and framework is easily generalizable such that it readily can be applied in other domains besides COVID-19 or cancer; by quickly loading new data-sets and associated feature-vectors.

8. References

[1] K. Jensen *et al.*, "Analysis of free text in electronic health records for identification of cancer patient trajectories," *Scientific Reports*, vol. 7, no. 1, art. no. 1, Apr. 2017, doi: 10.1038/srep46226.

[2] S. A. Murray, M. Kendall, K. Boyd, and A. Sheikh, "Illness trajectories and palliative care," *BMJ*, vol. 330, no. 7498, pp. 1007–1011, Apr. 2005, doi: 10.1136/bmj.330.7498.1007.

[3] "The Danish Cancer Society," *International*. https://www.cancer.dk/international/about-the-danish-cancer-society/ (accessed 09th August, 2020).

[4] "WHO Coronavirus Disease (COVID-19) Dashboard." https://covid19.who.int (accessed 09th August, 2020).

[5] G. Umefjord, K. Hamberg, H. Malker, and G. Petersson, "The use of an Internet-based Ask the Doctor Service involving family physicians: evaluation by a web survey," *Fam Pract*, vol. 23, no. 2, pp. 159–166, Apr. 2006, doi: 10.1093/fampra/cmi117.

[6] G. Umefjord, H. Sandström, H. Malker, and G. Petersson, "Medical text-based consultations on the Internet: A 4-year study," *International Journal of Medical Informatics*, vol. 77, no. 2, pp. 114–121, Feb. 2008, doi: 10.1016/j.ijmedinf.2007.01.009.

[7] A. K. Kushwaha and A. K. Kar, "Language Model-Driven Chatbot for Business to Address Marketing and Selection of Products," in *Re-imagining Diffusion and Adoption of Information Technology and Systems: A Continuing Conversation*, Cham, 2020, pp. 16–28, doi: 10.1007/978-3-030-64849-7_3.

[8] A. K. Kushwaha and A. K. Kar, "Micro-foundations of Artificial Intelligence

Adoption in Business: Making the Shift," in *Re-imagining Diffusion and Adoption of Information Technology and Systems: A Continuing Conversation*, Cham, 2020, pp. 249–260, doi: 10.1007/978-3-030-64849-7_22.

[9] A. K. Kushwaha, A. K. Kar, and P. Vigneswara Ilavarasan, "Predicting Information Diffusion on Twitter a Deep Learning Neural Network Model Using Custom Weighted Word Features," in *Responsible Design, Implementation and Use of Information and Communication Technology*, Cham, 2020, pp. 456–468, doi: 10.1007/978-3-030-44999-5_38.

[10] A. K. Kushwaha, S. Mandal, R. Pharswan, A. K. Kar, and P. V. Ilavarasan, "Studying Online Political Behaviours as Rituals: A Study of Social Media Behaviour Regarding the CAA," in *Re-imagining Diffusion and Adoption of Information Technology and Systems: A Continuing Conversation*, Cham, 2020, pp. 315–326, doi: 10.1007/978-3-030-64861-9_28.

[11] S. Ebadollahi, J. Sun, D. Gotz, J. Hu, D. Sow, and C. Neti, "Predicting Patient's Trajectory of Physiological Data using Temporal Trends in Similar Patients: A System for Near-Term Prognostics," *Amia Annual Symposium*, vol. 2010, pp. 192–196, 2010.

[12] "Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients | Nature Communications." https://www.nature.com/articles/ncomms5022 (accessed 09th August, 2020).

[13] X. Ji, S. A. Chun, and J. Geller, "Predicting Comorbid Conditions and Trajectories Using Social Health Records," *IEEE Transactions on NanoBioscience*, vol. 15, no. 4, pp. 371–379, Jun. 2016, doi: 10.1109/TNB.2016.2564299.

[14] O. Frunza, D. Inkpen, and T. Tran, "A Machine Learning Approach for Identifying Disease-Treatment Relations in Short Texts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 801–814, Jun. 2011, doi: 10.1109/TKDE.2010.152.

[15] C. Lousteau-Cazalet *et al.*, "A decision support system for eco-efficient biorefinery process comparison using a semantic approach," *Computers and Electronics in Agriculture*, vol. 127, pp. 351–367, Sep. 2016, doi: 10.1016/j.compag.2016.06.020.

[16] C. C. Yang and T. D. Ng, "Analyzing and Visualizing Web Opinion Development and Social Interactions With Density-Based Clustering," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1144–1155, Nov. 2011, doi: 10.1109/TSMCA.2011.2113334.

[17] C. Manning, P. Raghavan, and H. Schuetze, "Introduction to Information Retrieval," p. 581, 2009.

[18] *An algorithm for suffix stripping*. 1980.

[19] J. A. Goldsmith, D. Higgins, and S. Soglasnova, "Automatic Language-Specific Stemming in Information Retrieval," in *Cross-Language Information Retrieval and Evaluation*, Berlin, Heidelberg, 2001, pp. 273–283, doi: 10.1007/3-540-44645-1_27.

[20] C. H. Porter, L. E. Lynch, J. A. Herrig, and R. J. Ziebol, "(54) DEVICE AND METHOD FORVASCULAR ACCESS," p. 60.

[21] S. E. Robertson and K. Spärck Jones, "Simple, proven approaches to text retrieval," University of Cambridge, Computer Laboratory, UCAM-CL-TR-356, 1994. Accessed: 10th August, 2020. [Online]. Available: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-356.html.

[22] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *Journal of the American Society for Information Science*, vol. 27, no. 3, pp. 129–146, 1976, doi: 10.1002/asi.4630270302.

[23] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, Jan. 2004, doi: 10.1108/00220410410560582.

[24] Kar, Arpan, "Applications of Machine Learning in Business," *Business Frontiers*, 24th July, 2020. .

[25] A. Kar, "Understanding Machine Learning and Artificial Intelligence and their effects on Financial Systems – Business Fundas.".

[26] "Classification of Diseases and their Treatments Using Machine Learning Approach - ProQuest." https://search.proquest.com/openview/423cca63369eb17808ce3e845e51b852/1?cbl=2029261&pq-origsite=gscholar (accessed 12th August, 2020).