Mortality Prediction via Physicians' Notes

Nirave Kadakia, Ameet Chaubal
Georgia Institute of Technology, Atlanta, GA

Abstract

Sepsis is the leading cause of death in the hospitals. The disease also adds a severe financial burden on the healthcare system being the leading contributor to the medical expenses. In this paper, we explore the possibility of mortality prediction using free text notes available for each patient. The free text notes are used to perform topic modeling. The topics thus modeled are then employed to generate features which using Clustering analysis offer the prediction capability.

1. Introduction

Sepsis is the most expensive condition to treat in the US hospitals with an expense of $23.7 billion in the year 2013. Of the 1.6 million+ per year cases of sepsis, 40% with severe sepsis do not survive. Additionally, there has been an increase in hospitalization in the last 10 years for adult patients.

Consequently, it will be immensely beneficial to have an early detection of sepsis. In fact, an early detection carries a cost of $3000/case as opposed to $32,000/case after admission into a hospital.

On admission into the hospital, several attributes via chart events and vital signs are already present for each patient. More attributes are generated continuously from various tests conducted for the patient. In this paper, we focus primarily on the notes written by various physicians. The aim of a machine learning based analysis of patient data is eventually to replicate the intuition and decision making of a physician. As such, what better place to start at than the free text notes generated by various physicians for a given patient.

Although "Sepsis" has been described for over 2000 years, there is no "gold standard" to define it conclusively. Many conditions or infections, if left untreated, can eventually lead to festering which can cause organ failure and septic shock. Now, intuitively, there is a close relationship between an ICU admission and infection leading to "Sepsis". And in 2016, Sepsis and Septic Shock have been defined as, "*A life-threatening organ dysfunction due to a dysregulated host response to infection*". We can exploit this implicit relationship by looking at the mortality data in the ICU to arrive at Sepsis diagnosis.

We plan to use the MIMIC-III openly available dataset developed by the MIT lab for Computational Physiology. This database contains health data for approximately 40,000 critical care patients from the Beth Israel Deaconess Medical Center collected over a period of 10 years.

We demonstrate the use of free text notes as features for each patient. These features can then be leveraged in further machine learning analyses for prediction.

1.1 Paper Outline

Section 2 describes relevant literature and previous analysis performed by scientists. Section 3 highlights the technologies used and the rationale behind their selection. It also outlines the sequence of steps used in the analysis along with evaluation criteria. Section 4 details the methods used in analysis. Section 5 describes the results of the experiment and depict various outcomes from the models in appropriate formats. Section 6 provides a commentary on the results, while section 7 describes the direction in which future research can be conducted. Final section concludes with the result and the overall theme of the paper.

2. Previous Research

There is a well-established and prevalent system used in the hospitals, called "qSOFA" or quick Sequential Organ Failure Assessment. A qSOFA score of 2 or greater indicates "Sepsis". However, this test takes into account just three indicators – Respiratory rate, systolic blood pressure and altered Mentation or mental activity. qSOFA considers low blood pressure (SBP≤100 mmHg), high respiratory rate (≥22 breaths per min), or altered mentation (Glasgow coma scale<15).

Ghassemi et al focused on the free text notes for the same analysis. They devised a range of three prediction regimes, in which the free text notes are used to buttress the baseline indicators for analysis. Additionally, the authors considered prediction under two separate timelines – 30 day and 1 year post discharge. The authors normalized each note to a 50-dimension vector and the notes were then aggregated on a 12-hour scale. With this, combined with baseline features, they were able to achieve impressive AUCs ranging from 0.85 to 0.77, for in-hospital, post-discharge mortality ranging from 30 days to 1 year.

DeSautels et al applied a machine learning system called "Insight" focused on easily obtained patient data such as the vitals. They were able to achieve a better classification score than qSOFA. The authors considered a window of 48 hours before admission to 24 hours after the admission. The study demonstrates the potential of vital signs in predicting sepsis and mortality if applied with intelligent modifications.

3. Approach

3.1 Big Data Technology Stack and Rationale

The analysis relies upon big data technologies, specifically YARN and Spark. This will be driven by Postgres, which will perform data extraction and preparation. The advanced machine learning analyses will be performed using the machine learning library ML-lib on Spark.

The rationale for using the big data technology is simply the desire to process an ever growing amount of data. Although the current analysis is based on the freely available MIMIC-III database, our objective is that this type of analysis is generic enough to be abstracted into a pattern and as such can be applied to similar privately available information as well.

Apache Spark leverages the power of a distributed cluster by fanning out the data storage and computation across hundreds or thousands of computers and cores. A common data structure which holds datasets in Spark is called a Resilient Distributed Dataset or *RDD*. It is assumed that a dataset is so large that it cannot fit on a single node. Consequently, it's broken up into chunks and distributed across the entire cluster of machines. A "*Partition*" is the most atomic chunk of data that can fit on a single node. An RDD is a collection of multiple partitions. Spark then schedules these partitions for computation or processing across the nodes of the cluster. Spark will be used for data transformation as well since it can utilize the power of a Hadoop cluster.

The attraction of ML-lib is that it is designed for Spark and as such works on RDD. The algorithms have been designed for distributed processing on Spark. The first step of the analysis will revolve around "topic modeling" out of the free text notes.

3.2 Latent Dirichlet Allocation

LDA is a very effective method for topic generation out of a corpus of document. LDA aims to generate a collection of topics, where each topic is a collection of words and an assignment of topics for each document based on probability. These probability distributions have an interesting property called, "*sparsity*", which penalizes the assignment of a word to a topic and a topic to a document. The belief is that a topic has only a very few words that truly representative of it. Similarly, a document truly has a very small number of topics, mostly a single one, which it manifests based on the words. A document has a propensity towards a small number of topics and a topic has a propensity towards a small number of words. A document only

has a small number of salient words and these words are forced to choose just ONE topic; similarly a topic is forced to have a small number of words based on their probability ranking. Thus these two goals are in conflict with each other. LDA aims to strike a balance between the two.

3.3 Clustering by K-Means

Clustering aims to group objects in such a way that objects in the same group are more similar to each other than objects from other groups. K-means is a centroid-based clustering algorithm, which aims to decrease the distance between each point represented by a feature vector and the centroid. Each point is thus assigned to a centroid with the smallest distance among all the centroids. Spark ML-lib provides a parallelized K-means implementation which we plan to use for our analysis.

We chose K-means over Gaussian mixture model because of its simplicity, ability to produce accurate results and the "certainty" desired of a point belonging to one of the clusters. K-means has a better running time and works well with larger number of dimensions. In case we choose to increase the number of topics, K-means will scale well. The results are easier to interpret being part of spherical clusters. Additionally, the number of clusters is small; especially in case of mortality, we plan to use two clusters which should hopefully coincide with a patient being *dead* or *alive*.

3.4 Flow

Each patient may have multiple entries in the database with unique notes. Each of these notes, represents a story at the given moment in time. The objective of our analysis is to cobble together a representative "*narrative*" based on the individual stories which can present a singular picture vis-à-vis Sepsis and Mortality.

We extract the "*NOTEEVENTS*" table from the MIMIC-III database, focusing on two fields – subject and notes.

This data is loaded into HDFS, the distributed file system employed in the Hadoop or YARN cluster. HDFS enables us to process data without worrying about the size. If the available data is truly large in size, our approach will still work without modification as it simply piggy-backs on the power of the YARN cluster which can be scaled horizontally by increasing the number of nodes in the cluster. On loading into HDFS, the data is automatically broken up into chunks called "blocks" based on the block size configured on the cluster. The block size usually ranges from 128 MB to even 1GB. Since the size of the "*NOTEEVENTS*" table is relatively small, we chose a block size of *128 MB*. With the cluster size available, this enabled us to extract optimal level of parallelism from the nodes.

Next, we run the LDA analysis on the input data stored in HDFS. This generates three output files – topic information, map of documents to topics and map of topics to document. The topic output can be used for general understanding, while the document to topics map is used for the next phase of analysis.

The "*top Topics per Document*" file is used for the next stage in the pipeline, which is the K-means analysis. The output of the K-means produces clusters, where each cluster should map to a corresponding identifier. The identifier for each document in this case is the "subject_id" or the patient.

The final stage in the pipeline is the performance measurement of the mapping described above. We employed the "*purity*" measure to evaluate the performance of the clustering step. Purity is an external evaluation criterion of cluster quality and is expressed by,

$$\boldsymbol{Purity} = \frac{1}{\boldsymbol{N}} \sum_{\mathrm{i=1}}^{\mathrm{k}} \mathbf{max_j} \; |c_i \cap t_j|$$

Where

N = number of data points,

k = number of clusters,

c_i = a cluster

t_j = classification with the max count for cluster c_i

4. Method

In the following section, we detail the steps as highlighted in the above pipeline.

4.1 Preparation:

We leverage the MIMIC-III database for our analysis. This database contains 26 tables distributed as zipped "CSV" files. Each table has a "*subject_id*" which refers to the patient under consideration and can be used to link with other tables. These files once downloaded and extracted are loaded into Postgres database. Postgres is chosen for it stability and flexibility in addition to being an open source and freely available product, as well as providing off the shelf integration with Spark. It can be installed on multiple operating systems and has a mature ecosystem of client tools. Special care needs to be taken while importing the "*free text notes*", primarily because the notes are enclosed in quotations and contain carriage return characters as well as can be quite long. Postgres makes this task easier by interpreting all these cases while importing the data.

As described above, each patient may have multiple records with notes attached to it. Each note can be treated as a document in of itself and the entire collection can be considered a "corpus". However, a note is truly linked to a patient and belongs to the patient. Consequently, we posit that all the notes belonging to a patient be consolidated together into a single document. This lends more meaning to the document as the entire text can be analyzed to belong to a singular "topic", rather than each individual note belonging to separate topics. This also decreases the amount of documents in the corpus and optimizes the LDA analysis.

In order to make the import into Spark efficient, we ensure that the data extracted has the "notes" field devoid of any "carriage returns". This ensures that each note is a unique and single record for Spark and can be distributed across the cluster cleanly and efficiently.

4.2 LDA Analysis

An important pre-requisites for the LDA analysis is a set of "*stop-words*". These words are removed from the documents before processing. There are various "*stop-words*" dictionaries around and can be used as a starting point. However, for the medical free text notes used by the physicians, additional words need to be added to this dictionary. This was an iterative process which involved running through the analysis, evaluating the topics, generating the new stop words to be included in the set and then re-running the analysis. After several such iterations, we were able to arrive at an equilibrium and finalize a stop-words list.

There are two optimizers available for the Spark implementation of LDA – 1. Online LDA which implements the *variational Bayes* algorithm, processing a subset of the corpus on each iteration. 2. *Expectation Maximization* which implements the "smoothed model" outlined by Asuncion et al. Currently, the EM implementation is the only one which has methods to extract the bi-directional mapping between topics and document and that's the one used in this paper.

Empirically, we found that in order to extract relevant topics, at least 70 iterations of the algorithm were necessary. We ran our analysis on 100 iterations to retrieve the topics. The maximum words per topic was 10 as experimentally that seems to make sense. The top topics for each document with corresponding weights was saved to HDFS for further analysis.

4.3 Clustering

We proceeded to perform clustering analysis on the top topics per document saved as described in previous section 4.2. K-means made the most sense in the mortality prediction, since the data naturally lends itself to 2 clusters and there was a certainty about the affinity to the cluster for each patient.

The LDA analysis produces a file which has each document and the top topics for it. These records are converted to twenty features for each subject or patient and then loaded using Spark. The features are reduced using Principal Component Analysis to ten before carrying out the K-means training and prediction.

The real labels are extracted from the "ADMISSIONS" table in Postgres and using the labels, "purity" metric is computed.

5. Experimental Results

We experimented with 5, 10 and 20 topics and beyond that we started noticing duplication of keywords. In fact, the number of topics can be further reduced and simplified. Salient findings of our analysis are highlighted through the images depicted below.

5.1 Initial Experimentation

Before considering free-text analysis, some initial exploratory results were obtained to determine if sepsis detection could potentially work with machine learning tools coupled with the MIMIC-III dataset. It is important to note that sepsis detection was based on the MIMIC-III's Admissions (within ADMISSIONS.csv) sepsis diagnosis, meaning that sepsis's diagnosis was obvious, and thus would skew results. The purpose of this exploration was not obtain hard numbers, but to explore the usefulness of the data and potential models.

ICD-9 codes within admissions data were used as the input data, and the diagnosis of sepsis was the output data. To ensure that the sepsis diagnosis was not a result of an earlier admission that was diagnosed later, or part of an obvious connection from a recent event, the input data excludes the last 90 data days immediately prior to a sepsis diagnosis.

The results are based on 4 differing machine learning classification models, with a 70/30 split between testing and training data.

Algorithm	Train accuracy	Train ROC	Test accuracy	Test ROC
AdaBoost Classifier	0.976	0.5	0.977	**0.5**
SVC	0.995	0.933	0.983	**0.870**
Gradient Boosting Classifier	0.977	0.509	0.977	**0.499**
Decision Tree	1.0	1.0	0.985	**0.873**

Table 1: Initial experimentation

ROC, or the receiver operating characteristic curve accuracy ratio, against the test data is the most important measurement for measuring effectiveness since the data is binary classification, and with the most patients not having sepsis, measuring positive rate against the false positive rate is important. So that is the primary method used for model effectiveness.

An SVM and Decision Tree provide the highest ROC information in terms of sepsis prediction. This makes sense as those that experience sepsis are more likely linearly separable than those that are relatively healthy. A decision tree is also intuitively useful as the diagnosis of sepsis is often based on previous ICD codes - in other words, it maps how a doctor could potentially determine sepsis. It is unclear why *AdaBoost* and *Gradient Boosting* return low ROC values, perhaps due to incorrectly tuned hyper-parameters, but for the purposes of these early stage explorations, knowing that some machine learning algorithms work is all that is required, and no further work was needed.

It is important to note that the accuracy and ROC, as mentioned earlier, is skewed and does not represent true Sepsis numbers, which is why this is relegated to early exploratory research. According to *DeSautels* et al, 11.3% of all patients had sepsis prevalence in MIMIC-III, while the diagnosis of sepsis in MIMIC-III according to the ADMISSIONS.csv was approximately 2.36%. This tells of two factors - the labeling of sepsis is difficult, and that the method used above only determines the "most obvious" sepsis cases and ROC would most likely reduce dramatically if all 11.3% of sepsis patients were identified. Correct classification of sepsis will be explored in the future.

The conclusion of these preliminary results is that it is possible to predict sepsis in some capacity within MIMIC-III, and we can expand this further. More importantly, this provides confidence that the free text analysis will prove valuable, with results being discussed below.

5.2 Purity

The free text experimentation purity metric was computed to be **0.883**.

5.3 Descriptive Statistics

The top 5 topics by patient contribution are as follows,

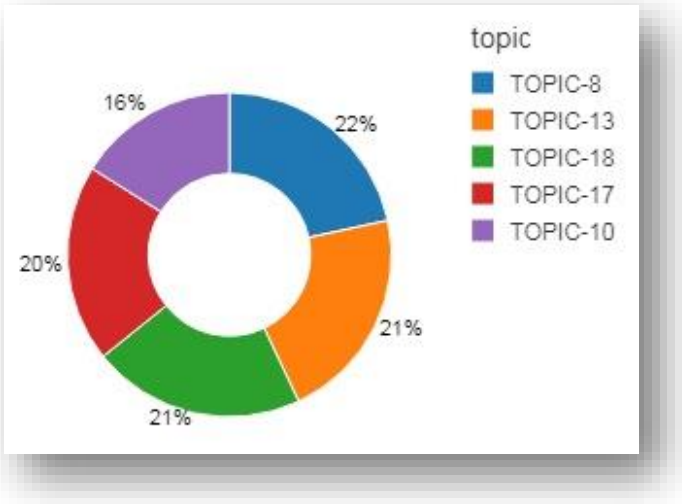


Figure 1: Top 5 topics by patient count

The patient count for topics is highlighted in the following image,

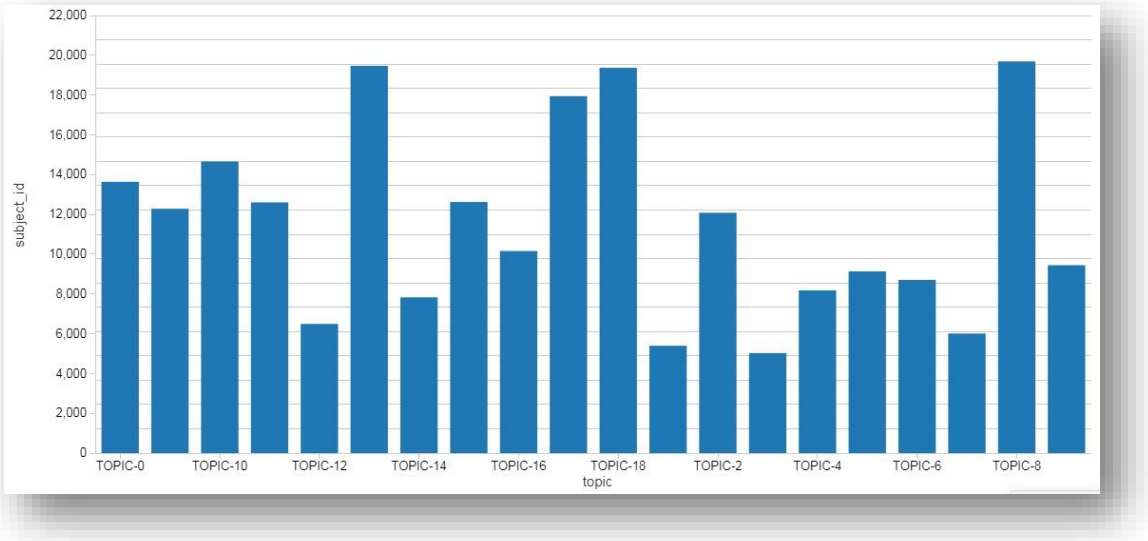


Figure 2: Topic count by patients

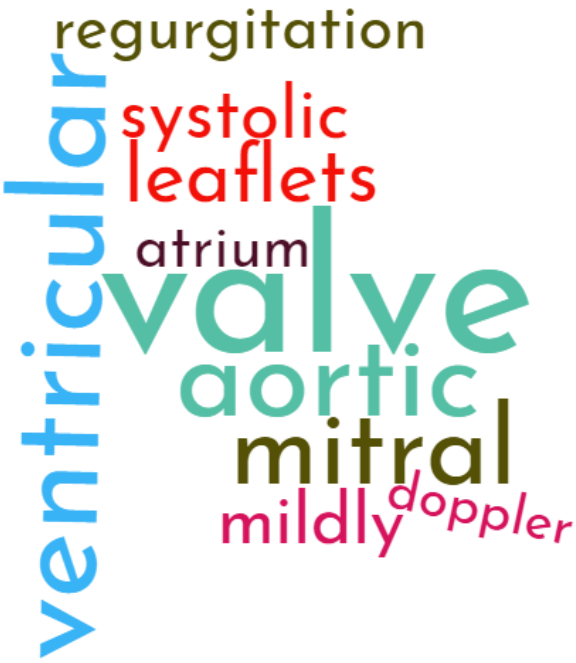
5.4 Word Cloud

The topics were used to generate word cloud; some of the interesting ones are as depicted below.

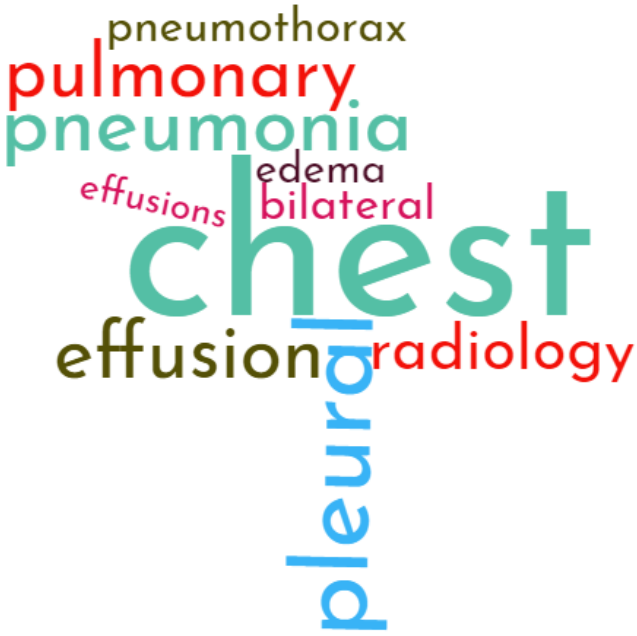
Topic-8

Topic-13

Topic-18



Topic-17



5.5 Topic and Keywords

The twenty topics are listed as follows,

Topic	Key Words
TOPIC-1	neuro, yellow, urine, foley, stool, intubated, social, draining, commands, nursing
TOPIC-2	cancer, lesion, metastatic, lesions, pelvis, abdomen, radiology, disease, tumor, lymph
TOPIC-3	Lasix, atrial, heart, cardiac, fibrillation, heparin, Coumadin, rhythm, failure, chronic
TOPIC-4	respiratory, failure, acute, sounds, pulse, prophylaxis, sputum, nutrition, intubated, ventilation
TOPIC-5	renal, failure, acute, sepsis, hypotension, dialysis, shock, levophed, infection, catheter
TOPIC-6	seizure, mental, abuse, alcohol, withdrawal, seizures, ativan, neuro, urine, acute
TOPIC-7	respiratory, secretions, nasal, received, sounds, bowel, lytes, chest, closely, ordered
TOPIC-8	trach, secretions, respiratory, sputum, suctioned, tracheostomy, sounds, fentanyl, propofol, sedation
TOPIC-9	chest, neuro, artery, stitle, foley, lungs, catheter, alert, oriented, radiology
TOPIC-10	hemorrhage, neuro, frontal, brain, subdural, cerebral, stroke, radiology, hematoma, carotid
TOPIC-11	fracture, spine, radiology, chest, trauma, fractures, cervical, posterior, injury, vertebral
TOPIC-12	chest, aortic, artery, coronary, valve, disease, bypass, aorta, graft, stenosis
TOPIC-13	liver, transplant, hepatic, ascites, cirrhosis, portal, bleed, bleeding, renal, radiology
TOPIC-14	insulin, sinus, chronic, denies, glucose, chest, acute, rhythm, renal, hypertension
TOPIC-15	infant, feeds, voiding, murmur, feeding, spells, spits, neonatology, sounds, retractions
TOPIC-16	chest, abdomen, radiology, abdominal, catheter, bowel, pleural, pelvis, effusion, obstruction
TOPIC-17	pulse, urine, stool, bleed, abdominal, allergies, bleeding, respiratory, rhythm, prophylaxis
TOPIC-18	chest, pleural, pneumonia, effusion, pulmonary, radiology, bilateral, pneumothorax, edema, effusions
TOPIC-19	valve, ventricular, aortic, mitral, leaflets, systolic, mildly, regurgitation, atrium, Doppler
TOPIC-20	sounds, extremities, respiratory, acute, rhythm, sodium, insulin, nutrition, drains, regular

Table 2: Topic and top 10 keywords

6. Discussion

The purity of the Clustering Analysis is very good, which indicates the power of the free text notes in predicting mortality. This should not be a surprise, since a physician is in the best position to leverage human experience, judgment and analytical capabilities after reviewing the patient at close quarters. Consequently, the notes implicitly contain most of the knowledge from the vital signs as well as test results.

K-means clustering produces the best result for clustering since k-means would naturally create centroids revolving around whether a patient is alive or dead..

Currently, we have not included any other features in our analysis. It is not known at the moment, whether addition of lab results would degrade or enhance the performance of prediction. This is one area where we feel that an expert opinion from a physician may be beneficial to ascertain the importance of certain features.

7. Future Work

We did not apply the technique of "*Stemming*" to the documents. Stemming algorithms aim to find a "morphological root" of a word. "*Porter's Algorithm*" is the most widely used mechanism for achieving the root. We think that stemming will dramatically reduce the vocabulary for the LDA analysis and provide a more condensed topic list.

We plan to modify the number of topics in LDA analysis. The duplication of key words suggests that the number of topics can be decreased further. It will be interesting to re-run the clustering for various topic numbers to analyze whether the purity numbers improve.

For prediction, we would like to evaluate the use of *Convolutional Neural Networks* as an alternative to K-means clustering.

It will be interesting to incorporate other features from the MIMIC-III database based on an expert opinion from an ER physician. Additionally, the features may help to eliminate non-septic mortality from the analysis. Currently, the prediction applies to general mortality with a tacit assumption of Sepsis being present for in-hospital deaths.

8. Conclusion

The paper demonstrates the power of free text notes in predicting in-hospital mortality. The notes through the feature creation offer excellent performance of **0.883** purity for mortality prediction. The main theme of the paper is that we perform topic modeling on free text notes using Latent Dirichlet Allocations. The LDA analysis as implemented on Spark has the potential to handle increasingly large amount of data on the backs of a large YARN cluster.

References

1. Desautels T, Calvert J, Hoffman J, Jay M. Prediction of sepsis in the intensive care unit with minimal electronic health record data: A machine learning approach. JMIR. 2016.
2. Ghassemi M, Naumann T, Finale DV, Brimmer N, Joshi R, Rumshisky A, Szolovits P. Unfolding physiological state: mortality modeling in intensive care units. ACM SIGKDD. 2014; 75-84
3. Blei, Ng, Jordan. Online Learning for Latent Dirichlet Allocation NIPS, 2010
4. Blei, Ng, Jordan: Latent Dirichlet Allocation. JMLR, 2003
5. Asuncion, Welling, Smyth, Teh: On smoothing and inference for topic models. UAI, 2009

Sources

Why does LDA work: https://www.quora.com/Why-does-LDA-work
Stemming: https://en.wikipedia.org/wiki/Stemming
Clustering evaluation: https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html