Automatic Topic Discovery of Online Hospital Reviews Using an Improved LDA with Variational Gibbs Sampling

Richard de Groof and Haiping Xu
Computer and Information Science Department
University of Massachusetts Dartmouth, Dartmouth, MA 02747, USA
Email: {rdegroof, hxu}@umassd.edu

***Abstract*—E-commerce websites such as Yelp.com, allow users to write online reviews of products and services, so new customers can have quick access to user experiences, covering everything from auto-repair to hospitals. However, a typical user may find it difficult to identify a topic of interest due to the overwhelming amount of review information. To deal with this issue, the Latent Dirichlet Allocation (LDA) model can be used to associate meaningful terms with text-based reviews, permitting keyword retrieval of individual documents. LDA is a powerful unsupervised learning approach, which has been widely used for topic modeling as well as in other related fields. A conventional implementation of LDA is through the Markov-Chain Monte-Carlo methodology, called Collapsed Gibbs Sampling (CGS). However, due to the usage of random numbers in the CGS approach, results from multiple trials on the same data are usually inconsistent. To avoid this tendency, we revise the conventional LDA approach using Variational Gibbs Sampling (VGS). VGS eliminates random numbers, and thus leads to consistent results as well as better performance. Our case study shows that our improved LDA can be used to automatically identify keywords and topics in online hospital reviews. Due to the usage of VGS, the accuracy of topic identification has been consistently improved.**

Keywords-hospital reviews; text mining; Gibbs sampling; latent Dirichlet allocation; topic modeling; big data

I. INTRODUCTION

With the proliferation of social media, there has been a growing opportunity for consumers to share their experiences. Websites such as Facebook, are filled with textual accounts of real people sharing their opinions of the quality of goods and services. This may provide an opportunity for an interested reader to find out what people think about a product. Similarly, Yelp.com allows a user to enter a query and location, bringing up postings and ratings. However, a drawback with Yelp.com is that there is no organization with regard to the topics of the review comments. As users may enter anything in their reviews and assessments, finding out only about a specific topic, such as waiting times for an urgent care center, may require sifting through hundreds of reviews before getting an accurate representation. Therefore, there is a critical need to find an effective way to classify the documents automatically, permitting users to find their topics of interest more easily.

There are four major types of machine learning approaches, namely supervised learning, semi-supervised learning, unsupervised learning and reinforcement learning [1]. Semi-supervised and supervised techniques require some degree of human intervention. In these approaches, labeled training and testing sets guide the process along, as in an intelligent process incorporating heuristics. Semi-supervised learning incorporates some amount of labeled data but a larger amount of unlabeled data. Reinforcement learning differs from standard supervised learning because it does not require correctly labeled input/output pairs; instead, an agent can learn its behavior based on feedback from the environment. Similar to reinforcement learning, an unsupervised learning technique also does not require any data labeling; thus, it provides some major advantages over supervised and semi-supervised learning approaches when it is costly to label data manually.

As a powerful topic-modeling methodology, Latent Dirichlet Allocation (LDA) uses unsupervised learning, which provides a probabilistic model for understanding text data [2]. A typical implementation of LDA is to use Collapsed Gibbs Sampling (CGS) [3], an application of the more general Gibbs Sampling technique. This approach is fast and efficient and widely used in LDA applications. A drawback of the approach, however, is that it relies on random numbers for its operation. Although the key idea with Gibbs Sampling is that a parameter of interest may be approximated using a sufficient number of samples, when the samples are generated as random draws from a distribution, the results of two trials of LDA using CGS that run on the same data may be inconsistent. This may not be what is desired from the application as businesses expect user experiences to be consistent. Thus, it is very important to seek improvements on the LDA approach in order to produce the same result from identical search queries.

In our proposed novel Variational Gibbs Sampling (VGS) approach, we eliminate random numbers to ensure consistent results between identical trials, as well as more accurate results. To assess our approach, we collected online reviews from Yelp.com, and used the LDA approach to derive keywords and topic distribution. Based on the distribution results, we further use *k*-means approach to clustering review sentences into major classes. We manually label the review sentences with topics, and calculate the classification accuracy of our VGS approach vs. the CGS approach. Though it is generally difficult to have very high accuracy using unsupervised techniques as using supervised ones, we show that with our improved LDA model, we are able to provide consistent results for our application of identifying major topics from online hospital reviews.

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

II. BACKGROUND AND RELATED WORK

A challenge in text categorization is the semantic distinction between identical word choices in a corpus, the same word occurring under different circumstances. Latent Semantic Analysis (LSA) has been used as a technique to overcome this difficulty [4]. By first decomposing the term-frequency matrix of a corpus into eigenvectors and eigenvalues (termed singular values in that work), and then reconstructing the matrix using a reduced set of singular values, a latent semantic space is produced, which represents a generalization of the original corpus. The relative occurrence of common words is emphasized across documents so that entities are more similar according to context as well as in the use of any particular word. Probabilistic Latent Semantic Analysis (PLSA) introduces a probabilistic generative process [5]. Using PLSA, when a writer creates a document, he first chooses a topic and then chooses a word to represent that topic. PLSA associates the latent topics with the co-occurrence of words within documents, modeled as a multinomial distribution, as in the repeated rolling of dice. In practice, the derivation of PLSA introduces a topic vector for each document/word combination. This is likely to result in overfitting, but the authors introduced a method to combat this tendency.

LDA is similar to PLSA in that it describes a probabilistic generative process. However, when selecting a topic described in a document, the topic is drawn from the Dirichlet distribution, a conjugate prior of the multinomial. The Dirichlet distribution can be tuned according to an input parameter (commonly referred to as α), by which more or less of its mass may fall towards some region of the simplex describing its other parameters (commonly referred to as θ). The formulation of LDA is intractable for direct inference. A commonly used approximation technique is called Gibbs Sampling [6]. Gibbs Sampling can be used to calculate a parameter of interest from a distribution given a set of samples from that distribution. It is a Markov-chain Monte-Carlo methodology in that it is iterative, stateless and utilizes random numbers. However, this tendency towards randomness may be undesirable.

Griffiths and Steyvers introduced CGS as an implementation of the LDA model [3]. It takes as input Dirichlet parameters α and β, a term-frequency matrix of a corpus and the desired number of topics, and outputs θ and φ, the distributions of topics over documents and words over topics, respectively. CGS has been implemented in different ways. For example, in [7] the authors combined a variational Bayesian inference approach with CGS; in [8], the authors proposed Fast Collapsed Gibbs Sampling (FCGS), a modification of CGS, which involved fewer operations to speed up the process. Some other work makes use of LDA in a semi-supervised manner. In [9] the authors proposed a novel method for text classification using LDA and semi-supervised learning. Their process proceeds in a loop, utilizing a small set of training labels to assess the quality of the LDA model and unsupervised classifications, and then shifting unlabeled data to labeled ones.

Other methods have also been proposed for automatic topic discovery using LDA. In [10] the authors used FCGS to generate the LDA parameters and Shannon information to extract keywords. Also, in [11], the authors proposed a text classification mechanism based on LDA. They incorporated sentiment analysis into the process to increase accuracy. The focus in that work was on subjective topics, so their results show an improvement in incorporating this information.

Another variation of LDA was used in [12], where tweet followings were mapped in an LDA-based model. The authors of that paper identified experts in microblogs and incorporated topic models from followers into information gathered about the subject. LDA has been used in [13] where the authors incorporated data commonly excluded from consideration as noise. They used emoticons in microblogs as consideration in topic mining. They argue that this information is readily available in this day and age, and is becoming more prevalent and should be incorporated into the state of the art.

Different from the above approaches, we propose a novel technique, called Variational Gibbs Sampling (VGS), which can be used to implement the LDA approach more effectively. Our case study shows that our approach outperforms LDA with CGS approach, and also produces consistent results between runs on identical data.

III. ANALYSIS OF HOSPITAL REVIEWS USING LDA

Figure 1 shows the overall approach to using LDA for topic identification and keyword retrieval. In our approach, reviews are first split into sentences and tokenized, forming documents of nouns and adjectives with stop-words removed. These documents are used as input in the improved LDA process incorporating VGS. The output of LDA is the matrices θ and φ. Through an operation on these matrices, we may derive the significant words (keywords) for each document. The matrices derived in this operation serve as the starting point for unsupervised clustering.

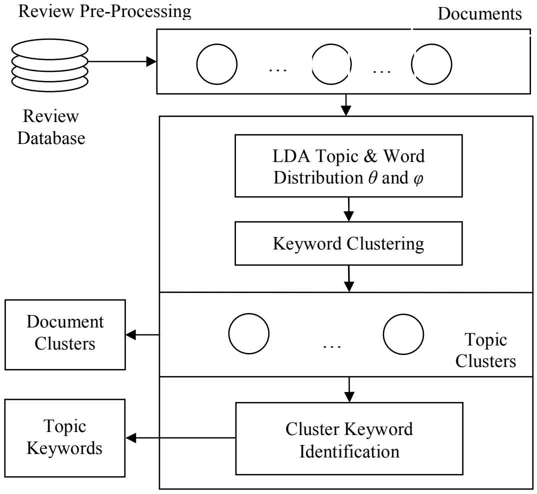

Figure 1. Topic identification using an improved LDA

In the clustering process, we use *k*-means to group the keywords into classes according to their significance. We then aggregate the values in these clusters into a cluster centroid by averaging the values for each word. By comparing the occurrence of words in documents with the centroids, we cluster the documents into related groups. Finally, we calculate the most frequently occurring terms to derive topic keywords, which are useful in retrieval of the clustered documents.

IV. LDA WITH VARIATIONAL GIBBS SAMPLING

A. *LDA Using Collapsed Gibbs Sampling*

The LDA model we adopted is depicted in Fig. 2. The boxes denoted as *D* and *W* represent repetitions at the document-level and the word-level, respectively. The random variable θ represents the underlying probability of the individual topics for a document, which is drawn from the Dirichlet distribution given Dirichlet parameter α. Random variable z is the latent topic drawn from the multinomial distribution given θ. The random variable w presents the word drawn from the multinomial distribution given z and φ, the distribution of words over topics, which is also drawn from the Dirichlet distribution, given Dirichlet parameter β.

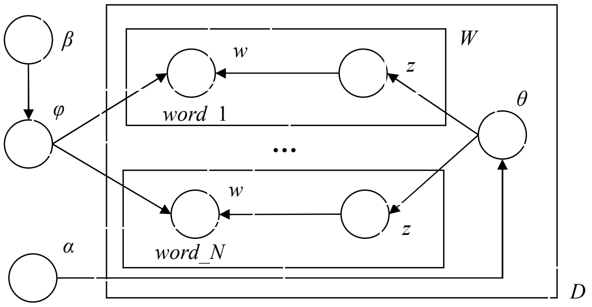

Figure 2. The LDA model for selecting words for a document

The Dirichlet parameters α and β are *K*-sized and *N*-sized vectors for word over topic and topic over document Dirichlet distributions, respectively. They could be used to tune the Dirichlet distributions, where the greater proportion of the total value of the vector in a part of the β vector associates greater probability with those corresponding topics, the same being true for the α vector and words.

According to this formulation, the complete joint probability of a corpus and assignment of topics to it can be defined as in (1).

$$P(W,Z,\theta,\varphi;\alpha,\beta)=\prod_{i=1}^{K}P(\varphi_i;\beta)\prod_{j=1}^{M}[P(\theta_j;\alpha)\prod_{t=1}^{N}[P(z_{j,t}\,|\,\theta_j)P(w_{j,t}\,|\,\varphi_{z_{j,t}})]] \tag{1}$$

where W is a corpus as defined by the counts of words occurring across documents; Z is the set of topic assignments to documents and words; M is the number of documents; N is the size of the vocabulary, i.e., the total number of words occurring across all documents; and K is the predefined number of topics. The semi-colons in (1) represent that the probability is calculated based on the given Dirichlet distribution parameters α and/or β.

As mentioned previously, LDA is a generative statistical model; therefore, sample data can be produced according to probabilistic distributions. The process of generating a document can be analogized to throwing a dice with some sizes of the faces, corresponding to topic distributions over documents. Once a topic is selected, the words can be determined using a separate set of dice corresponding to word distribution over each of the topics. By first rearranging terms and integrating over θ and φ, with the integral of the Dirichlet distribution being equal to one, the joint probability in (1) of a particular word and topic assignment given the stationary point of all other topic assignments is proportional to the term as defined in (2).

$$P(Z_{(m,n)}=k,Z_{-(m,n)},W;\alpha,\beta)\propto (n_{m,(.)}^{k,-(m,n)}+\alpha_k)\frac{(n_{(.),n}^{k,-(m,n)}+\beta_n)}{\sum_{r=1}^{N}(n_{(.),r}^{k,-(m,n)}+\beta_r)} \tag{2}$$

where $n_{j,r}^{i,-(m,n)}$ denotes the count of the i-th topic for the j-th document and the r-th word; (.) indicates that the value represents the sum across that parameter (i.e. across all documents or all words); $-(m,n)$ indicates that the contribution from document m and word n have been excluded from that count; and $Z_{-(m,n)}$ indicates all other topic assignments excluding the one currently being sampled. Like a random walk, the next state is assigned according only to the current state. This translates to the stationary state of all other variables, all other topic assignments, excluding the one in question. In the original CGS formulation, the next topic is assigned randomly. It certainly involves random numbers, although, as we will see shortly, there is a certain degree of regularity in the assignments. Note that (2) can be used as the CGS update equation.

CGS proceeds in a three-nested loop for a predefined number of iterations over documents and words. By maintaining counts of topics associated with documents and words, the probability of each topic is calculated and stored in a vector. Also, the present topic for a document/word combination is maintained as the starting point for each iteration. The selection of the topic which is assigned to that document and that word combination in CGS is done according to a random draw from the calculated topic vector, a process which approximates the underlying distribution. In CGS, the counts of topics assigned to documents and words are first populated with initial values, which may be far from the expected values. The reason that the process is called Markov-Chain Monte-Carlo is because a Markov-Chain is a series of states in which the next state may be generated from only the preceding one. It is Monte-Carlo because it uses random numbers. Algorithm 1 shows this process, but with the selection of the updated topic being done according to the TS-VGS algorithm presented in Section IV.B.

Algorithm 1: Variational Gibbs Sampling (VGS)

Input: *word* is an $M \times N$ matrix with the number of times word w has appeared in document m; α and β are Dirichlet parameters; T is the number of times to repeat sampling.

Output: θ is an $M \times K$ matrix representing the distribution of topics over documents; φ is an $N \times K$ matrix representing the distribution of words over topics.

```
Let topic be $M \times N$ matrix representing the topic for a pair of
  document $m$ and word $w$.
Let nw and nd be $N \times K$ and $M \times K$ matrix containing the
  number of times word $w$ and document $m$ being assigned to
  topic $k$, respectively.
Let nwsum be a $K$-sized vector representing the number of
  times topic $k$ has been assigned to a word.
Let ndsum be a $M$-sized vector representing the number of
  times document $d$ has been assigned a topic.
for $i = 1$ to $T$
    for $m = 1$ to $M$
        for each $w$ in word occurring in document $m$
            $n$ = index of word $w$ in vector word[$m$]
            $k$ = topic[$m$][$n$]
            nw[$n$][$k$]--; nd[$m$][$k$]--, nwsum[$k$]--; ndsum[$m$]--
            curTopic = TS-VGS(nw, nd, $\alpha$, $\beta$, $m$, $n$)
            nw[$n$][curTopic]++; nd[$m$][curTopic]++
            nwsum[curTopic]++; ndsum[$m$]++
            topic[$m$][$n$] = curTopic
for $m = 1$ to $M$
    for $k = 1$ to $K$
        $\theta[m][k] = (nd[m][k] + \alpha[k]) / (ndsum[m] + \sum_j \alpha[j])$
for $n = 1$ to $N$
    for $k = 1$ to $K$
        $\varphi[n][k] = (nw[n][k] + \beta[n]) / (nwsum[k] + \sum_j \beta[j])$
return $\theta$ and $\varphi$
```

In Algorithm 1, the counts of the current sample are first decremented. This allows for the sampling of the current state given all other states. The probabilities of the topic assignments *topic*[*m*][*n*] are calculated at each iteration for each document *m* and word with index *n*. Different from CGS that makes a random selection of topic from the vector of topics, this algorithm calls method TS-VGS to determine the current topic for a given document and a word, and update the counts and assignments accordingly.

B. Variational Gibbs Sampling

Before presenting the topic selecting algorithm using VGS, we first show how CGS works. The major idea of CGS is to associate frequently occurring words with topics. Having no prior knowledge about the underlying topic distribution, one may first initialize the topic/document and topic/word counts in a pseudo-random fashion, assigning topics to documents and words as they appear in the corpus. Therefore, in the beginning, there will be a large degree of disorder in the topic vector *pr_topic*, a vector of topic probability produced for each document/word combination. An early iteration of the algorithm produces a normalized topic vector, which may look as follows:

[0.001][0.25][0.15][0.001][0.55][0.048]

Using CGS, the vector *pr_topic* is first cumulated left to right by iterating across the vector and summing each value with all prior values, which results in the following vector:

[0.001][0.251][0.401][0.402][0.952][1.00]

Then, a uniform random value between 0 and 1 is calculated and multiplied by the last value, which is guaranteed to contain the complete probability. The vector is iterated over and, for the first value that is greater than this random value, its index is the one selected as the topic assignment for that document and word combination.

We can see that the CGS algorithm tends to select the topics with the most significant values. In the cumulated vector above, there is a 95.2% certainty that topic 5 is selected, and it is guaranteed that otherwise, topic 6 will be. If the same word is associated with a different topic in another iteration of the algorithm, then the probability of that topic will be higher at that index on the next iteration, and thus, the topic assignments will be more likely to be moved in that direction. Practically, however, the use of random numbers allows for any topic choice to be made. In the example above, 25% of the time, the second topic will be selected and 40% of the time the third topic will be selected. In fact, the distribution is spread largely across the 2nd, 3rd, 4th and 5th topics. The topic selection algorithm should reflect this situation, and pick a point in the middle of these four consistently according to the distribution.

In contrast, VGS works by associating the majority mass of the distribution with a non-random point so that the underlying distribution is accurately represented. Lines 7 to 19 in Algorithm 2 show this process. First, the topic vector is accumulated in two directions, from the left and right towards the point of the highest value. In the example above, the cumulated vector would look like:

[0.001][0.251][0.401][0.402][1.0][0.048]

Then we calculate the sum across this cumulated vector, and call it *pr_topicSum*. For the above vector, it is 2.103. Finally, by iterating from left to right, the values are summed incrementally, denoted as *runningSum*. The index at which the incremental sum represents the majority of the vector (e.g., *runningSum*/*pr_topicSum* > 0.5) will be the topic index returned. According to this procedure, the above example returns topic index 4.

The summing of the vector represents the mass at that point. Consider the vector as a function varying in amount along the list of the topics. Summing across the vector is similar to taking an integral of the function from the beginning of the vector to each index of the vector. This allows us to find the majority mass of the vector.

This process might better represent the underlying distribution as the values to the left of the highest point represent a significant proportion of the total mass of the vector. In a different scenario, when the mass of the vector were located at the opposite end of the vector, for example,

[0.048] [1.0] [0.402] [0.401] [0.251][0.001]

Algorithm 2: Topic Selection Using VGS (TS-VGS)

Input: *nw* is $N \times K$ matrix containing the number of times word *w* has been assigned to topic *k*; *nd* is $M \times K$ matrix containing the number of times document *m* has been assigned to topic *k*; α and β are Dirichlet parameter*s*; *m* and *n* are the current document index, and the current word index, respectively.

Output: topic index for Gibbs Sampling update equation (2)

```
Let nwsum be a K-sized vector representing the number of
    times topic k has been assigned a word by summing nw over
    all words per topic.
Let pr_topic be a K-sized vector of topic probabilities.
for $k = 1$ to $K$
    $pr\_topic[k] = (nw[n][k] + \beta[n])$ /
            $(nwsum[k] + \sum_n \beta[n]) * (nd[m][k] + \alpha[k])$
Normalize pr_topic such that $\sum_i pr\_topic[i] = 1$
Let apex be the highest value in pr_topic;
    apexIndex = the index in pr_topic of apex
for counter = 2 to apexIndex   // cumulate pr_topic
    $pr\_topic[counter] =$
        $pr\_topic[counter] + pr\_topic[counter-1]$
for $counter = K - 1$ to apexIndex
    $pr\_topic[counter] = pr\_topic[counter] +$
        $pr\_topic[counter+1]$
Let pr_topicSum be the sum over pr_topic
// find the point of the greatest mass in pr_topic
runningSum = 0.0
for counter = 1 to K
    $runningSum = runningSum + pr\_topic[counter]$
    if ($runningSum/pr\_topicSum > 0.5$) return counter
return K
```

then in this case, the total mass would be the same but the topic assignment becomes 3, shifted just right of the highest point in the direction of greatest proportionate mass. The distributions of mass in these examples imply that, for iterations of other documents containing the same word, these topics would also represent significant values. It might be beneficial if the process could recognize this, and accumulate values along the topics so that different documents will have a more consistent topic assignment, which we hope represents the same underlying topic.

Having the topic assignment done as in Algorithm 2, subsequent topic assignments will tend to acknowledge this topic selection for the (*m*, *n*) pair. In addition, topic assignments for other documents containing the same word *n* will do so in the same way. On the other hand, through a random assignment, CGS does the similar thing for a proportion of the topic mass towards one end of the vector or another, which is used for topic assignment at that point. However, CGS may diverge from this tendency because, ultimately, the selection of any topic is possible.

V. Automatic Topic Identification Using Improved LDA

The results of LDA using VGS are the matrices φ and θ, i.e., the distributions of words over topics and topics over documents, respectively. The most significant terms appearing in a document can be calculated as in (3).

$$SignificantTerms_d = \theta_d \times \varphi^T \qquad (3)$$

where θ_d is a vector of topic probabilities for document *d*, and φ^T is the transpose matrix of topic probabilities for words appearing in the corpus.

The resulting vector of (3) is an *N*-sized vector, for which the highest entries are the most significant words. The matrix φ provides the significance of each word according to topic. Sentences composed of similar combinations of words will be associated with a particular topic distribution. φ will be distributed according to the relative frequency of the word for those topics. In this way, equation (3) provides a distribution representing the significance of words according to topic for a single document. For each document, the most significant words will have the highest value in the *N*-sized vector. Then, we use *k*-means clustering to group all the vectors produced by (3). The significant terms reflective of a particular topic will be associated with a certain cluster. We aggregate each cluster individually, averaging across columns to produce a generalization of the word significance. Lastly, for each sentence, we compare the words that occur with each cluster aggregate, summing significance for each occurring term, to determine the greatest similarity. Documents most similar with a particular sequence of term significances (having the highest sum across topic significances) are placed in that cluster. In this way, the documents are aggregated according to significant terms. We demonstrate in Section V that this approach is very effective in clustering documents according to hidden topics. In addition, we may extract the significant terms per cluster, which can be used as keywords that are associated with a hidden topic.

VI. Case Study

In this section, we demonstrate that our novel approach to LDA using VGS can be applied effectively to online hospital reviews for clustering review sentences by topics and identifying keywords related to each topic. Furthermore, we show that our VGS approach typically performs better than CGS in our trials.

A. Data Processing

We collected 92 hospital reviews from Yelp.com, and split them into 192 sentences. Each sentence is then tokenized, identifying Parts of Speach (PoS) using Apache OpenNLP [14]. Tokenizing the sentences can be beneficial in our approach because informative words, such as nouns and adjectives, can be obtained readily. Note that in our approach, we focus on nouns and adjectives, where nouns are particularly useful because they may serve as keywords. For example, the following sentence contains a noun, “staff”, and an adjective, “friendly”, which is often used to describe a staff:

> “The staff were all very friendly.”

Therefore, tokenization can be used to eliminate "stop words" (words we do not care about and are commonly occurring across topics). In our approach, words like "the", "and" and "or" are not adjectives or nouns, and thus are eliminated. Here is an example of a review, labeled as Review 39 in our data set. The review can be split into two sentences, each of which contains at least one noun or at least one adjective.

> "The secretary at the front desk signed me in very quickly. I literally waited 10 minutes and was called into the waiting room."

B. *Unsupervised Clustering Using LDA*

In order to demonstrate the effectiveness of our approach, we performed experiments on the 192 sentences using LDA with both CGS and VGS methodologies. The resulting matrices θ and the transpose of φ were multiplied to produce a term-significance matrix. This matrix was then clustered using k-means with varying numbers of centroids, and each cluster was aggregated column-wise to produce a single generalization of word significance per cluster. Each document was compared with each of the cluster aggregations to accumulate word significance of occurring words and cluster the documents into related clusters. Finally, to compute the accuracy of the two approaches according to cluster fidelity to a particular topic, we manually identified the topics according to the terms which occurred in each cluster. The clusters produced were composed in the following way, with the above cluster containing one sentence from the example review 39:

**************Cluster: 1 *****************

Words in sentence	**Review number**
minutes, hour, dr, office, norm	32
hour, comfortable, half, less	24
doctor, room, while	65
minutes, office, dr., wait, nurse, doctor	37
minutes, room, waiting	39
minutes, info, intake	48
minutes, home, good, 10am	1
hour, dr., x-rays	28

In the above example, each line shows the terms occurring in a sentence, which served as a document in our trials of LDA. One may observe some coherence in the cluster. For example, the word "minutes" is highly indicative of the topic "waiting time," a frequent source of conversation in descriptions of hospitals. Other clusters were composed of varying amounts of significant keywords for other topics.

With a larger number of centroids input into k-means in our approach, there were a larger number of clusters output. Tables I and II show the number of clusters produced by running k-means on the matrices produced by each method. Notice that the number of clusters tended to increase with both methods but was more stable when using VGS. This indicates that there was a higher degree of cohesion in the results of that method. Tables I and II also show that, with the VGS approach, there is a high degree of consistency between the numbers of clusters as the number of centroids used with k-means increases. Even at the highest number of centroids, the sizes of the clusters were largely consistent but also appropriate for the proportions of each topic.

TABLE I. Number Of Clusters Produced by Kmeans (CGS)

Num. Centroids	**Num. Clusters**	**Avg. Sentences per Cluster**	**Std. Dev. (Sentences per Cluster)**
6	6	32	18.18
8	8	24	8.82
10	10	19.2	3.54
12	10	19.2	3.54
14	11	17.45	6.04
16	13	14.77	8.2
18	13	14.77	8.15

TABLE II. Number Of Clusters Produced by Kmeans (VGS)

Num. Centroids	**Num. Clusters**	**Avg. Sentences per Cluster**	**Std. Dev. (Sentences per Cluster)**
6	6	32	14.97
8	8	24	7.81
10	10	19.2	5.31
12	10	19.2	5.31
14	10	19.2	5.10
16	11	17.45	6.96
18	11	17.45	6.96

C. *Result Analysis*

Once we have associated labels with the series of terms, we can assess the accuracy of the two methodologies. The following is an example of a labeled cluster:

******************Cluster: 9 ****************

Staff : *friendly, girls, desk, front, super, willing*
Staff : *doctor, questions, intelligent*
Staff : *doctor, informative, times, polite*
Staff : *staff, professional, desk, front, nurses*
Staff : *test, nice, drug, lady, young*
Staff : *front, nice, woman*
Staff: *doctor, opinion, daughter, unprofessional*
Staff : *girls, desk, nice, i*
Staff : *highly, facility*
Staff : *place, neighborhood, available*
Staff : *facility, kids*
Staff : *desk, front, secretary*
WaitingTime: *wait, appointment, online, scheduler, perfect*
WaitingTime: *exam, form, standard*

We treated our manual labeling as the ground-truth, and identified K = 6 underlying topics from the collected review sentences with the following amounts:

Medications: 22, WaitingTime: 39, Staff: 58, Billing: 24, Cleanliness: 10, Null: 39

The "Null" topic includes sentences that do not describe any particular aspect of the hospital. A sentence such as "So glad to have such a great urgent care in Arlington Heights," does not contain a topic. It only describes a general feeling about the hospital as a whole. We included the "Null" topic in our case study to demonstrate the discriminative power of our methodology. The "Null" topic is trickier to identify manually than the rest since it tends to be composed of less informative and more varied terms.

Having the ground-truth labels, we may measure the accuracy of the clusters generated in our experiment according to (4).

$$Accuracy = \frac{\sum_{j} Entry_{k,j} \mid Entry_{k,j} \geq Entry_{i,j} \text{ for } i = 1..K}{\sum_{j}\sum_{i} Entry_{i,j}} \qquad (4)$$

where $Entry_{k,j}$ is the number of sentences in the j-th cluster for topic k, which contains the majority of sentences in cluster j among all topics. Therefore, we calculate the accuracy as the total number of majority sentences (i.e. across all clusters) divided by the total number of sentences. Table III shows these results for both of the CGS and VGS approaches with different numbers of centroids. As is shown in Table III, the results are almost always better for VGS as compared with CGS.

TABLE III. ACCURACY OF CGS AND VGS

Num. Centroids	CGS	VGS
6	0.609	0.604
8	0.635	0.656
10	0.651	0.703
12	0.651	0.703
14	0.646	0.708

Table IV further shows the number of sentences per cluster for 12 centroids and either method. From the table, we can see that even with a large number of centroids, both VGS and CGS maintained a largely similar number of clusters. Note that a topic may appear as the majority in more than one cluster. This is not a problem because, as we show in Section VI.D, the keywords can be used to associate the separate clusters of the majority topic.

Additionally, to demonstrate the consistency of VGS vs. CGS, we present the results from six trials of the two methodologies as in the preceding experiment, classified with 14 centroids. Table V shows the results. It is clear that the results of CGS may vary significantly between runs. This is the problem due to the random process – though it seems to make good associations which are useful in classification, the experimental results show that it just as often makes poor associations.

Table VI presents the majority topic for each cluster produced by the approach using both methodologies and 12 centroids. The results show both approaches work equally well in topic matching.

TABLE IV. NUMBER OF SENTENCES PER CLUSTER

Cluster No.	CGS	VGS
1	17	15
2	14	15
3	27	14
4	17	15
5	17	33
6	21	21
7	21	19
8	21	21
9	16	21
10	21	18

TABLE V. ACCURACY FOR SIX TRIALS OF CGS VS. VGS

Run No.	CGS	VGS
1	0.594	0.708
2	0.604	0.708
3	0.630	0.708
4	0.573	0.708
5	0.615	0.708
6	0.578	0.708

TABLE VI. MAJORITY TOPIC FOR EACH CLUSTER (12 CENTROIDS)

Cluster No.	CGS	VGS
1	Null	Billing
2	Medication	Staff
3	Waiting Time	Staff
4	Billing	Null
5	Staff	Waiting Time
6	Staff	Medication
7	Staff	Staff
8	Null	Staff
9	Waiting Time	Null
10	Staff	Staff

D. *Keyword Identification and Topic Discovery*

All results were obtained using LDA with 10 as the predefined number of topics. Thus, it makes sense that VGS would converge to roughly 10 clusters despite an increasing number of centroids. With these many clusters generated from the VGS approach, and six underlying topics as the ground truth, one could imagine that further clustering approach would be necessary to make the VGS approach more effective in a completely unsupervised manner. In the following, we describe how the word occurrences within the clusters may help to aggregate them further.

Having the clustered documents, we could associate them with meaningful terms. By aggregating the terms in the clustered sentences, we may find the significant ones as those most frequently occurring. Table VII shows the most significant words occurring per cluster and the majority topics for the results of our process run using VGS and 14 centroids. Note that the word "staff" occurs in four out of five of the clusters assigned to that topic. However, as we may aggregate the separate clusters according to common word occurrences, the methodology is resilient to variations regarding the number of topics LDA is run with, and also the number of centroids used in our classification scheme.

TABLE VII. SIGNIFICANT WORDS ACCORDING TO CLUSTER

Cluster	**Word 1**	**Word 2**	**Word 3**	**Word 4**	**Word 5**
Null 1	*care*	*urgent*	*friendly*	*wait*	*insurance*
Null 2	*service*	*brand*	*office*	*customer*	*heck*
Billing 1	*service*	*brand*	*office*	*bill*	*billing*
Billing 2	*years*	*dr*	*insurance*	*doctor*	*compression*
Medication	*prescription*	*place*	*doctors*	*clean*	*quick*
Staff 1	*friendly*	*doctors*	*dr.*	*office*	*staff*
Staff 2	*doctor*	*staff*	*office*	*test*	*front*
Staff 3	*staff*	*great*	*friendly*	*office*	*clean*
Staff 4	*reception*	*comments*	*professional*	*time*	*minutes*
Staff 5	*professional*	*staff*	*wait*	*office*	*insurance*
Waiting Time	*minutes*	*hour*	*wait*	*doctor*	*time*
Null 3	*care*	*insurance*	*friendly*	*wait*	*co-pay*

Table VII also shows that we may associate frequently occurring terms as keywords to a set of documents. Generally, the keywords retrieved in this way are telling as to the topic of the cluster. A person might query the word "staff" and get documents describing the hospital staff, for example. A person might also search for "wait" and "time" and get back many sentences that discuss the waiting time of the hospital in question.

VII. CONCLUSIONS AND FUTURE WORK

We have introduced and demonstrated a revised Gibbs sampling methodology, called VGS, which does not require the calculation of random numbers. VGS calculates the maximum variance of the underlying distribution, separating the corpus into distinct clusters. Using this approach, we were able to classify sentences and identify keywords associated with sets of documents. In our trials for online hospital reviews, VGS performed better than CGS overall.

A key benefit of the VGS approach is its consistency. The results of identical queries will be the same in each occurrence. Thus, the provider of a keyword searching mechanism can guarantee the results be identical for various users. This may make the method more applicable in commercial situations, where this is generally desirable.

For future work, we intend to bolster the methodology by incorporating other machine learning techniques. While approximating the underlying distribution may produce consistent results, a reinforcement learning mechanism may possibly produce better accuracy [15]. In addition, we will validate our approach on large volumes of online review data using big data analysis, and further study on efficient ways of text mining for related scenarios [16][17].

REFERENCES

[1] E. Alpaydin, *Introduction to Machine Learning*, Second Edition, The MIT Press, Cambridge, MA, USA, 2010.

[2] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, March 2003, pp. 993-1022.

[3] T. L. Griffiths and M. Steyvers, "Finding Scientific Topics," *Proc. National Academy of Sciences*, vol. 101. Suppl. 1, April 6, 2004, pp. 5228-5235.

[4] N. E. Evangelopoulos, "Latent Semantic Analysis," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 4, no. 6, November 2013, pp. 683-692.

[5] T. Hofmann, "Probabilistic Latent Semantic Indexing," *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR99)*, Berkely, CA, USA, August 15-19, 1999, pp. 50-57.

[6] I. Yildirim, "Bayesian Inference: Gibbs Sampling," *Technical Note*, Department of Brain and Cognitive Sciences, University of Rochester, August 2012.

[7] W. Jiang, "Study on Identification of Subjective Sentences in Product Reviews Based on Weekly Supervised Topic Model," *Journal of Software*, vol. 9, no. 7, July 2014, pp. 1952-1959.

[8] I. Porteous, D. Newman, A. Ihler, A. Asuncion,P. Smyth and M. Welling, "Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation," *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Las Vegas, NV, USA, August 2008, pp. 569-577.

[9] M. Pavlinek and P. Vili, "Text Classification Method Based on Self-Training and LDA Topic Models," *Expert Systems with Applications*, vol. 80, September 2017, pp. 83-93.

[10] S. Jing and W. Li, "Topic Discovery Based on LDA Model with Fast Gibbs Sampling," *Proc. of the IEEE International Conference on Artificial Intelligence and Computational Intelligence (AICI'09)*, Shanghai, China, November 7-8, 2009, pp. 91-95.

[11] Y. Yu, M. Lingfei and W. Jian, "Identifying Topic-Specific Experts on Microblog," *KSII Transactions on Internet & Information Systems*, vol. 10, no. 6, June 2016, pp. 2627-2647.

[12] F. Colace, M. De Santo, L. Greco and N. Paolo, "Text Classification Using a Few Labeled Examples," *Computers in Human Behavior*, vol. 30, January 2014, pp. 689-697.

[13] F. Huang, S. Zhang, J. Zhang and G. Yu, "Multimodal Learning for Topic Sentiment Analysis in Microblogging," *Neurocomputing*, vol. 30, August 2017, pp. 144-153.

[14] Apache, "Apache OpenNLP Developer Documentation," *Manual*, Version 1.8.3, The Apache Software Foundation, 2017. Retrieved from https://opennlp.apache.org/docs/1.8.3/manual/opennlp.html on September 1, 2017.

[15] M. Camara, O. Bonham-Carter and J. Jumadinova, "A Multi-Agent System with Reinforcement Learning Agents for Biomedical Text Mining," *Proc. of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '15)*, Atlanta, Georgia, September 9-12, 2015, pp. 634-643.

[16] S. Boytcheva, G Angelova, Z. Angelov and D. Tcharaktchiev, "Text Mining and Big Data Analytics for Retrospective Analysis of Clinical Texts from Outpatient Care," *Cybernetics and Information Technologies*, vol. 15, no. 4, 2015, pp. 58-77.

[17] S. Jiang, X. Qian, J. Shen, Y. Fu and T. Mei, "Author Topic Model-Based Collaborative Filtering for Personalized POI Recommendations," *IEEE Transactions on Multimedia (TMM)*, June 2015, vol. 17, no. 6, pp. 907-918.