A Framework to Automatically Discover Posts about Depression in LymeNet using Machine Learning Techniques

Aditya Kalyan Jayanti

April 2020

Abstract

Lyme disease is one of the most common vector-borne illnesses in the United States, and it costs the healthcare industry billions of dollars each year. The CDC (center for disease control) estimates that there are about 300,000 cases each year. Due to the lack of knowledge on the progression of this disease, many patients seek social media channels to share health-care questions about Lyme disease, meet others with similar symptoms, and understand the progression of this disease. Depression is a significant contributor as a result of chronic illness and affects approximately 70% of the patients who do not consult doctors during the early stages of depression. The goal of this research is to expand the prevailing study by developing a machine learning system to identify posts about depression in LymeNet. In this paper, we use topic model algorithms to determine abstract topics in a collection of records and conduct a study on expressions of information concerning Lyme disease. We also develop and execute an algorithm to measure the similarity between topic models.

1 Introduction

Acute conditions such as common cold, high fever, etc., are severe and may occur abruptly, but they often persist for a short span [25]. A medical practitioner can treat several acute ailments without complex medicines and hospitalization. But, acute ailments not treated in time can lead to chronic conditions, these are persistent, long-lasting, and takes physicians a long duration to accurately diagnose it. The CDC [3] estimates that six in ten adults in the United States have at least one type of chronic disease as a result, it costs the country billions of dollars in health care annually [1]. For example, it can be difficult to diagnose Lyme disease as most symptoms are common with other illnesses like fever, headache, and fatigue, and hence it is necessary to develop new tools that can help doctors make an early diagnosis.

Because of the complex nature of this disease, patients require more than a scheduled doctor visit as it is often unrecognized and undiagnosed. They seek

online resources to understand and manage their condition and for finding others with similar symptoms [24]. These online support groups can influence how people understand their illnesses. In a recent study by the PRC (Pew research center) [4] on the social life of health information, they discovered that the number of people using online resources to look up health information increased from 25% in 2000 to 61% in 2009 [4]. However, the downside of depending on online support groups is that information can be inaccurate, inconsistent, misleading, and contradictory. While Mankoff et al. [11] discuss the effect of conflicting information for people having prolonged or recurring Lyme disease symptoms in greater detail, this research identifies latent topics to determine depression among individuals with Lyme disease symptoms.

Depression is a common complication of several chronic illnesses and often worsens symptoms of an existing illness [20]. Hence, it is crucial to develop techniques for recognizing depression in an individual during the early stages. Topic modeling is an effective way to discover the underlying theme in large-scale data [7], it is an effective machine learning tool to reduce the multidimensional data obtained from social media to a small number of topics.

We use machine learning to understand how people use social media to make sense of a controversial disease. The contributions of this paper are:

- Collect and process Medical Question data from LymeNet forums.
- Perform analysis driven by topic modeling on this data, and evaluate topic models using coherence and perplexity to determine the optimal topic size.
- Develop a new approach to evaluate topic models using similarity metrics.
- Assess topics within models using topic model visualization methods.

The report has been organized as, section 2 provides a brief overview of existing work in the areas of topic modeling, social media analysis, and unsupervised machine learning algorithms. In section 3 a subset of the data used for this research is described followed by a description of attributes. Section 4 provides a detailed description on the methods used in this research. Results and inferences are drawn in section 5. Limitations of this research are presented in section 6. Conclusions and future work are drawn in section 7.

2 Related Work

Due to the increase in the number of people using social media to share information, several types of research have appropriated this platform to identify depression by gathering data on social media. While the focus of our research is on identifying topics and associating them with depression, Shen et al. [21] describes a framework that incorporates distinct features to unveil discrepancy between depressed and non-depressed users on social media. They compared the performance of a multimodal framework against naïve bayes, multiple social networking, and Wasserstein dictionary learning [21], the multimodal framework

combined with dictionary learning resulted in the most effective model for the timely detection of depression. This timely detection of depression [15] limits the accelerating pace of deteriorating conditions and allows for earlier diagnoses before the onset of symptoms.

Many researchers have earlier studied online health support groups for chronic-illnesses that include cancer, hearing loss, autism, and mental health. But, when several online sources present competing viewpoints, it affects participants in understanding the disease. Mankoff et al. [11] presented a comprehensive study on the effect of conflicting online information for patients with recurring Lyme disease. The authors built a model using data obtained from online websites, medical articles, and survey participants to classify them into dominant, minority, and alternative types. This analysis supported patients to make informed decisions through reliable information.

The massive large-scale information in social media has prompted researchers to extract this data to predict stock prices and election results. Yet, predicting depression using social media is still in the early stages of development. Tsugawa et al. [24] presents an instance that relates to the concept described in this paper. The authors analyze data derived from Twitter to predict depression in Japanese users by extracting several features and activity history. Further, they computed frequencies of words in a tweet to classify between negative and positive words using SVM (support vector machine) and performed topic modeling using LDA. But, in contrast to depression associated with a chronic disease, the goal of Tsugawa et al. [24] was to predict depression to aid medical professionals in early diagnoses or for self-diagnosis.

LDA is a popular topic modeling method to discover themes in a large collection of unstructured textual information [2]. However, current research does not support the selection of an optimum number of topics. Zhao et. al [26] have introduced a new approach that uses the RPC (rate of perplexity change), to determine the number of topics in LDA. Perplexity is an informational theory that determines the effectiveness with which a statistical model describes a dataset. The disadvantage of this method is that it converges to different local optima for the same dataset. In contrast, the RPC approach results in an accurate and valid conclusion when tested against three distinct datasets of varying data types.

The disadvantage of LDA is the complexity involved in large scale data analysis despite using optimization techniques like variational inference [8]. An alternative method is using batch variational inference [2], but due to frequent modification of the dataset on each iteration, it can lead to expensive operations and can sometimes be computationally impractical for very large datasets. The central problem for topic modeling was to fit models with a more comprehensive corpus. Online LDA based on stochastic optimization overcomes this drawback by executing faster on large datasets. It also strikes the appropriate equilibrium between extracting knowledge gained from a particular topic and exploring new topics. Hoffman et al. [8] studied the performance of two algorithms online variational inference for LDA, and batch variational inference on a dataset consisting of 3.3 million Wikipedia articles. While the former algorithm runs with

constant memory requirements, the latter produces similar results but outperforms the former in terms of rate of convergence regardless of the size of the dataset.

Another shortcoming of LDA is its scalability, particularly for deployment in industrial applications. The largest deployment of LDA used about [13] a thousand computers to process more than eight million documents in ten hours. Alexander et al. [22] proposed a distributed memory caching method that parallelizes processes in a multicore configuration. While this method necessitates a multicore system, it reduces the number of computational resources required (e.g., thousands of computers) for processing. A parallelized LDA implementation can help improve the time taken to generate LDA models and improve accuracy in determining the topic size. An application of highly scaleable LDA is user recommendation in social media. Marco et al. [16] describe this application to automatically discover users' interests by comparing topic distributions using KL (Kullback Leibler) divergence. In our approach, we apply a similar distance metrics to compare topic distributions for analyzing an optimal number of topics.

3 Data

The data required for this research is web scrapped from an online message board known as LDN (Lyme disease network) or LymeNet.org, this website contains general information about the diagnosis, symptoms, and treatment of Lyme disease contributed by individuals affected by this disease. This online discussion board consists of six active forums 'Seeking a Doctor,' 'Activism,' 'Medical Questions,' 'General Support,' 'Off-Topic,' and 'Computer Questions'. For this research, we extract the 'Medical Question' forum data consisting of 153,173 records and topics into a JSON (javascript object notation) file. The JSON file is then imported into a Mongo database called 'lymeDiseaseDB'.

A subset of this data described in figure 1, shows that each document in the collection consists of these basic attributes '_id,' 'text,' 'userID,' 'postOrComment,' 'dateTime,' and 'topic'. Besides these basic attributes, the collection contains document topics and topic entropy, computed through the processing of the text field.

Figure 1: Medical Question Data in MongoDB Compass.

4 Methods

The link to my Bitbucket repository: `https://bitbucket.org/tripperroc/lymenetanalysis/branch/aditya-jayanti`. In this research, we used the open-source gensim library to perform unsupervised topic modeling and natural language processing. LDA is a topic modeling algorithm that generates topics based on word frequency. It consists of a three-level Bayesian structure that transforms each item in the corpus into a finite mixture [2]. The following subsections describe the process of generating topic models, comparing several topic models with similarity metrics and determining the optimal number of topics.

4.1 Data collection and processing

The process starts with collecting documents from Lymenet forums using web scraping tools followed by data cleaning to eliminate stop words. Following this, a word tokenizer split strings based on space and punctuation. There are two approaches to convert a word to its root form they are, stemming and lemmatization. In this study, we performed lemmatization using the NLTK library to reduce each word in the list to its root form. The benefit of lemmatization is, it considers the context of the text and converts it into a meaningful root form, whereas stemming may often lead to spelling errors [19]. For example, the words 'play' and 'played' are derived from the root 'play'. The next step involves generating a histogram that represents the number of words in the text. Followed by converting the document into a bag-of-words format using doc2bow, we transformed this representation into train and test data. Since we

don't perform predictions train, and test data contain the same set of values.

4.2 Training the LDA model

Topic modeling is frequently used in text mining to discover hidden semantic structures in a corpus. The underlying generative process for LDA is described in [2] in more detail. A parameter list created with arbitrary values initially determines the number of a topic model. We then train the LDA model by initializing the hyperparameters corpus, id2word, num_topics, eval_every, chunksize, and passes. On each iteration, the parameter value in the parameter list is assigned to num_topics. The following hyperparameters remain constant throughout the training process:

- corpus: Requires a stream of document vectors or a sparse matrix, it is assigned with the training dataset generated in step 1.
- id2word: Used for debugging and to determine vocabulary size. It is initialized with a dictionary generated in step 1 using the gensim library.
- passes: Represents the number of epochs (iterations) through the corpus during training. In this case it is set at 40.
- eval_every: Setting this parameter very low, slows down the speed of training the model by a factor of 2. In this case, it is initialized to 10.
- chunksize: Determines the number of documents used in each pass, the value remains constant at 3000.

Since LDA is an unsupervised technique, the number of topics that exist in the corpus is unknown before training the model. We can visualize the keywords for each topic in a topic model, and its importance using the method [9] lda_model.print_topics(). On each iteration, we save the LDA generated model and evaluate them using similarity metrics.

4.3 Evaluating a topic model

In this study, we use topic coherence and perplexity to evaluate topic models. Topic coherence measures the degree of similarity between high scoring words in a topic in a topic model [23]. Considering a set of words w_1 to w_n, the pairwise scores between two words w_i and w_j [18] are computed as:

$$Coherence = \sum_{i<j} Score(w_i, w_j)$$

This measure helps to distinguish between semantically related topics (e.g., search volume, keyword search) and statistically inferred topics (e.g., concluding from a random sample). The two types of coherence metrics (score) used in this paper are:

- UCI metric: It uses a pairwise scoring function based on PMI (pointwise mutual information) introduced in [14]. It works by computing word frequencies in a sliding window method over an external corpus (e.g., Wikipedia) [23].

$$Score_{uci}(w_i, w_j) = log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

- UMass metric: It uses a pairwise scoring function based on document co-occurrence introduced in [12]. It works by computing the fraction of documents containing w_i and w_j and the number of documents containing w_i [23] in the corpus used to train the LDA model.

$$Score_{umass}(w_i, w_j) = log \frac{D(w_i, w_j) + 1}{D(w_i)}$$

In our experiments, we applied the intrinsic UMass metric to estimate coherence and evaluate the topic model. A desirable model will generate coherent topics with high scores. On each iteration of the parameter value, we record the coherence value and plot them on a graph to discover the optimal number of topics. Perplexity is another evaluation method for topic models. In contrast to coherence, we find the smallest value that optimizes perplexity [17]. But, optimizing for perplexity does not lead to human interpretable topics, and hence coherence is a more reliable metric [17].

$$Perplexity(Test_{data}) = e^{-\frac{Log\ Likelihood}{count\ of\ tokens}}$$

4.4 Determining topic size using similarity metrics

In previous sections, we determined two metrics perplexity and coherence in determining the number of topics (topic size). This section discusses a novel method to discover topic size by comparing the similarity between topic models. We developed and implemented an algorithm to measure the similarity between topic models, section 4.4.1 discusses this in more detail. This process started by iterating over each topic model and determining the topic distribution pairwise. The distance metrics used in this algorithm are:

- Jensen-Shannon distance: It measures the similarity between two probability distributions and is based on KL divergence [5].

- Kendall-Tau distance: It measures the correlation between two arrays of the same shape. The values closer to 1 indicate strong agreement and values closer to -1 indicate strong disagreement [6].

4.4.1 Measuring similarity between two models of the same size

Given two topic models, $T^{(1)}$ and $T^{(2)}$, with topics $(t_1^{(1)}, \dots t_k^{(1)})$ and $(t_1^{(2)}, \dots t_k^{(2)})$ we describe a general method to measure the distance between them. Let $d_{\mathcal{X}}(t_{j_1}^{(i_1)}, t_{t_2}^{(i_2)})$ be a divergence measure between two probability distribution. We are particularly interested in $d_{\mathcal{X}} = d_{JS}$ (Jensen-Shannon distance) and $d_{k\tau}$ (Kendall tau distance).

Algorithm 1 LDA_{dist}
Input: topic models $T^{(1)} = (t_1^{(1)}, \dots, t_m^{(1)})$ and $T^{(2)} = (t_1^{(2)}, \dots, t_m^{(2)})$, *ordered by topic distribution*

Let Q be a priority queue of pairs of topics in $T^{(1)} \times T^{(2)}$, ordered by $d_{\mathcal{X}}(\cdot, \cdot)$
$T \leftarrow \emptyset$
while $Q \neq \emptyset$ **do**
  $(t_i^{(1)}, t_j^{(2)}) \leftarrow \arg\min_{(t_k^{(1)}, t_l^{(2)}) \in Q} d_{\mathcal{X}}(t_k^{(1)}, t_l^{(2)})$
  Append $(t_i^{(1)}, t_j^{(2)})$ to T
  Remove from Q all pairs containing either $t_i^{(1)}$ or $t_j^{(2)}$.
end while
Return the Kendall tau distance between $(1, \dots, m)$ and $(\pi(1), \dots, \pi(m))$, where π is defined such that, for all $i \in \{1, \dots, m\}$, $(t_i^{(1)}, t_{\pi(i)}^{(2)}) \in T$

4.4.2 Evaluating the algorithm

We evaluate the algorithm through the following steps: In the first step, we construct ten topic models using LDA. Followed by ordering topics in each topic model by their probability distribution. The third step consists of comparing the topics in each model pairwise by applying the Jensen-Shannon distance metric. On each iteration, we append the pair of topics associated with the minimum distance to a list and ultimately return the Kendall-Tau distance between them. We can graph the data to observe the variation between topic models and define the number of topics similar to coherence and perplexity.

5 Results

The LDA algorithm executed over the Medical Question Data with several hyperparameter values resulted in optimizing coherence, perplexity, and generated the optimal number of topics required to associate depression with Lyme disease. The initial run included the following hyperparameters:

- The number of topics, k = 140
- The number of passes = 40

- The value of alpha, α = "auto"

The term α in LDA represents the document-topic density, a higher value indicates that documents contain more number of topics and a lower value indicates that documents contain fewer topics. But, in this case, "auto" indicates that the model learns the priors from the data or corpus [2]. The goal is to identify a range of topics that optimize coherence and perplexity.

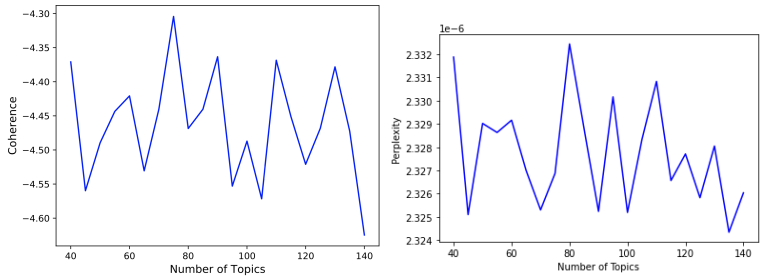

Figure 2: Plot for coherence and perplexity for 140 topics

The model is executed for 140 topics to identify regions that optimize coherence and perplexity. Figure 2 shows the results obtained for 140 topics. In this plot, the number of topics observed before it peaks to the highest coherence value was between 70 and 80. Similarly, the number of topics corresponding to the lowest perplexity value was between 130 and 140.

Building on these results, the second run included the following hyperparameters:

- The number of topics, k = 140
- The number of passes = 20
- The value of alpha, α = "auto"

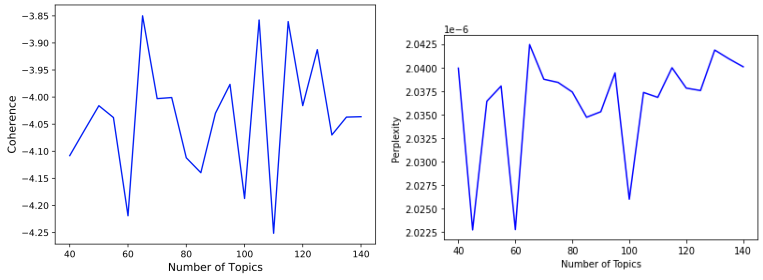

Figure 3: Plot for coherence and perplexity for 140 topics

Figure 3 shows the results obtained for 140 topics. In this plot, the number of topics observed before it peaks to the highest coherence value was between

60 and 70. However, unlike the previous execution the lowest perplexity value was between 60 and 70.

Determining the topic size is a challenging task since it widely differs from the type of data processed. The results generated in the above graphs correspond to several iterations with varying hyperparameters. We have also implemented the novel algorithm described in section 4.4.1, using this algorithm, we can test the topic size using similarity measures. Section 4.4.2 describes the experimental setup of generating and comparing ten topic models, and it included the following hyperparameters:

- The number of topics, k = 100
- The number of passes = 40
- The value of alpha, α = "auto"

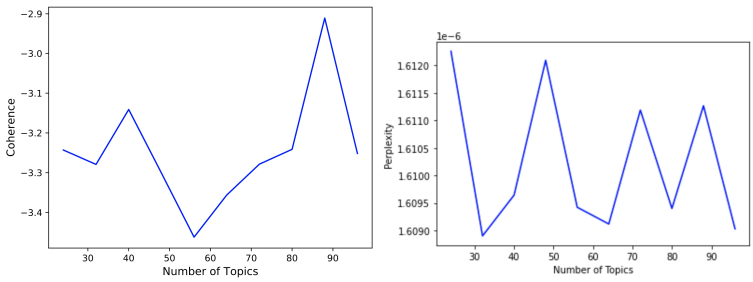

Figure 4: Plot for coherence and perplexity for 100 topics

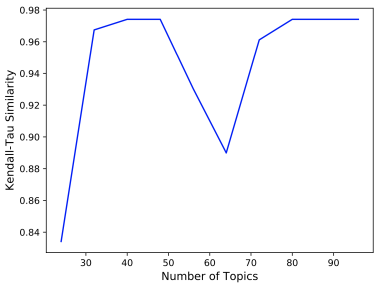

Figure 5: Plot for Kendall-Tau for 10 topics models

In this execution, we represented coherence, perplexity, and Kendall-Tau in figure 4 and figure 5.

- Coherence: The number of topics corresponding to the first peak was between 35 and 40. But, the number of topics corresponding to the second peak was between 85 and 90.

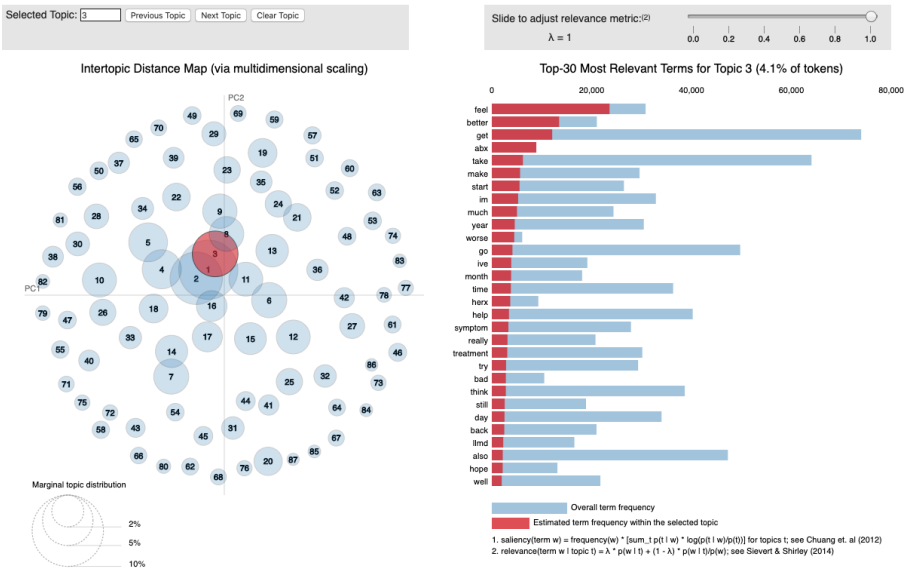

Figure 6: Topic Model Visualization

- Perplexity: The number of topics corresponding to the first peak was between 30 and 35. But, the number of topics corresponding to the second peak was around 90.
- Kendall-Tau: The number of topics corresponding to the first peak was between 35 and 40. But, the number of topics corresponding to the second peak was greater than 80.

Although topics between 80 to 90 have good coherence scores, it may contain repeated keywords, hence the increase in coherence measures. The algorithm provides a distinct approach to precisely recognize the optimal topic size. By inferring from the above information, we observe that the number of topics is between 30 and 40. Another method to interpret the information contained in each topic model is through pyLDAvis [10]. It is a Python library that generates an interactive web-based visualization to interpret topics in a topic model. Since topic models are usually unlabelled, a benefit of understanding these topics is to define their relation to a distinct behavior or character.

Figure 6 describes the result of executing pyLDAvis on a topic model, it consists of two panels:

- The left panel describes the prevalence of each topic and its relation to each other. These topics are represented as circles and its prevalence is determined by the circle's area.
- The right panel describes a bar chart representing the most useful terms for the currently selected topic.

6 Discussion

The popular topic modeling algorithms include LSA (latent semantic analysis), HDP (hierarchical dirichlet process), and LDA (latent dirichlet allocation). Among these, LDA is the most widely accepted topic modeling algorithm with better performance. In this study, we answered research questions related to reading topic models, comparing topics between several topic models and developed a unique approach to ascertain the optimal number of topics by comparing the similarity between different topic models. The limitation of this research is in its processing time because, of machine capacity, we used the single-core LdaModel for training. We can overcome this limitation by implementing LdaMulticore [27] this method, uses all CPU cores to parallelize, and speed up training.

7 Conclusion

In this study, we implemented the LDA topic model algorithm using Medical Question data and experimented with various hyperparameters to discover the optimal number of topics. In addition to coherence and perplexity, we also developed another approach for evaluating topic models. The topic model evaluation brings structure to unstructured data. Further research into topic model evaluation revealed that correlation with human interpretability plays an important role in deciding whether it is a good or bad indicator of the quality of topics. Perplexity is a poor indicator of topic quality, due to its anti-correlation with human judgment [17]. However, coherence assesses topic models better in contrast to perplexity. We developed an algorithm that reduces the dependency on human judgment by comparing the probability distribution of each topic between topic models. Using this algorithm, we determined the topic size to be between 30 and 40. The limitation of this method is, it takes several hours to complete execution and construct the respective graphs. In the future, sharded corpus and multicore topic modeling can reduce the time taken for topic modeling. Also, we can generate more topic models (e.g., twenty topic models) with varying values of k and compare the number of topics associated with the peak values of coherence, perplexity, and Kendall-Tau.

References

[1] E. R. Adrion, J. Aucott, K. W. Lemke, and J. P. Weiner. Health care costs, utilization and patterns of care following lyme disease. *PloS one*, 10(2), 2015.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] CDC. *Chronic Diseases in America*, 2020 (accessed February 21, 2020).

[4] P. R. Center. *The Social Life of Health Information*, 2020 (accessed February 22, 2020).

[5] T. S. community. Jensen-shannon distance (metric), April 2020. [Online; accessed 20-Apr-2020].

[6] T. S. community. Kendall-tau distance (metric), April 2020. [Online; accessed 20-Apr-2020].

[7] D. Greene, D. O'Callaghan, and P. Cunningham. How many topics? stability analysis for topic models. In T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 498–513, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[8] M. Hoffman, F. R. Bach, and D. M. Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[9] S. Kapadia. Evaluate topic models: Latent dirichlet allocation (lda), April 2019. [Online; accessed 20-Mar-2020].

[10] B. Mabey. pyldavis, March 2020. [Online; accessed 20-Mar-2020].

[11] J. Mankoff, K. Kuksenok, S. Kiesler, J. A. Rode, and K. Waldman. Competing online viewpoints and models of chronic illness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 589–598, New York, NY, USA, 2011. Association for Computing Machinery.

[12] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*, pages 262–272. Association for Computational Linguistics, 2011.

[13] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10(Aug):1801–1828, 2009.

[14] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[15] C. W. M. Ng, C. H. How, and Y. P. Ng. Major depression in primary care: making the diagnosis. *Singapore medical journal*, 57(11):591, 2016.

[16] M. Pennacchiotti and S. Gurumurthy. Investigating topic models for social media user recommendation. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, page 101–102, New York, NY, USA, 2011. Association for Computing Machinery.

[17] Q. Pleplé. Perplexity to evaluate topic models, May 2013. [Online; accessed 20-Mar-2020].

[18] Q. Pleplé. Topic coherence to evaluate topic models, May 2013. [Online; accessed 20-Mar-2020].

[19] S. Prabhakaran. Lemmatization Approaches with Examples in Python. `https://www.machinelearningplus.com/nlp/lemmatization-examples-python/`, 2019. [Online; accessed 23-Apr-2020].

[20] B. Saha, T. Nguyen, D. Phung, and S. Venkatesh. A framework for classifying online mental health-related communities with an interest in depression. *IEEE Journal of Biomedical and Health Informatics*, 20(4):1008–1015, July 2016.

[21] G. Shen, J. Jia, L. Nie, F. Feng, C. Zhang, T. Hu, T.-S. Chua, and W. Zhu. Depression detection via harvesting social media: A multimodal dictionary learning solution. In *IJCAI*, pages 3838–3844, 2017.

[22] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Proceedings of the VLDB Endowment*, 3(1-2):703–710, 2010.

[23] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961. Association for Computational Linguistics, 2012.

[24] S. Tsugawa, Y. Kikuchi, F. Kishino, K. Nakajima, Y. Itoh, and H. Ohsaki. Recognizing depression from twitter activity. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3187–3196, New York, NY, USA, 2015. Association for Computing Machinery.

[25] WebMD. *Coping With Chronic Illnesses and Depression*, 2020 (accessed February 22, 2020).

[26] W. Zhao, J. J. Chen, R. Perkins, Z. Liu, W. Ge, Y. Ding, and W. Zou. A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC Bioinformatics*, 16(13):S8, 2015.

[27] R. Řehůřek. Multicore lda in python: from over-night to over-lunch, September 2014. [Online; accessed 27-Apr-2020].

A Appendix

Timeline	Tasks	Hours
Weeks 1 - 4	Set up development environment, Human subjects certification, Draft of abstract, Background reading.	42
Week 5	Resolve errors in code base, Reproduce results for a fixed number of topics, Write up on related work, Further reading on Topic Modeling.	16
Week 6	Run LDA on 100 topics for 40 epochs and compute loss function, Research on comparing topic similarity and matching topics using KL divergence, Jensen-Shannon distance, and Cosine similarity.	18
Week 7	Generate an interactive plot to visualize topic models, Write up on introduction, Research on measuring distance between two LDA models.	12
Weeks 8 - 9	Run several iterations of LDA on different topic values until convergence, Identify regions that optimize coherence and perplexity, Document methods and execution results in report.	24
Weeks 10 - 12	Research on topic distribution and usage (gensim library), Generate topic distribution for 10 topic models, Develop an algorithm to determine topic size by comparing similarity between topic models.	33
Weeks 13-14	Compare difference between topic matching using Kendall-Tau, Generate plots to visualize topic size using the new topic model evaluation method.	22
Weeks 15	Integrate code for the algorithm into the code base, Document results and graphs, Revise report.	11
Final Week	Total time spent on independent study.	178