Yoga-Veganism: Correlation Mining of Twitter Health Data

Tunazzina Islam
Department of Computer Science
Purdue University, West Lafayette
IN-47907, USA
islam32@purdue.edu

ABSTRACT

Nowadays social media is a huge platform of data. People usually share their interest, thoughts via discussions, tweets, status. It is not possible to go through all the data manually. We need to mine the data to explore hidden patterns or unknown correlations, find out the dominant topic in data and understand people's interest through the discussions. In this work, we explore Twitter data related to health. We extract the popular topics under different categories (e.g. diet, exercise) discussed in Twitter via topic modeling, observe model behavior on new tweets, discover interesting correlation (i.e. Yoga-Veganism). We evaluate accuracy by comparing with ground truth using manual annotation both for train and test data.

CCS CONCEPTS

• **Human-centered computing → Social media**; • **Computing methodologies → Information extraction**; **Topic modeling**; **Factorization methods**.

KEYWORDS

Correlation mining; Topic modeling; Social media; Health

ACM Reference Format:
Tunazzina Islam. 2019. Yoga-Veganism: Correlation Mining of Twitter Health Data. In *Proceedings of 8^{th} KDD Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM@KDD'19), August 04–08, 2019, Anchorage, Alaska, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The main motivation of this work has been started with a question "What do people do to maintain their health?"– some people do balanced diet, some do exercise. Among diet plans some people maintain vegetarian diet/vegan diet, among exercises some people do swimming, cycling or yoga. There are people who do both. If we want to know the answers of the following questions– "How many people follow diet?", "How many people do yoga?", "Does yogi follow vegetarian/vegan diet?", may be we could ask our acquainted person but this will provide very few intuition about the data. Nowadays people usually share their interests, thoughts via discussions, tweets, status in social media (i.e. Facebook, Twitter, Instagram etc.). It's huge amount of data and it's not possible to go through all the data manually. We need to mine the data to get overall statistics and then we will also be able to find some interesting correlation of data.

Several works have been done on prediction of social media content [6], [5], [8], [19], [23]. Prieto et al. proposed a method to extract a set of tweets to estimate and track the incidence of health conditions in society [18]. Discovering public health topics and themes in tweets had been examined by Prier et al. [17]. Yoon et al. described a practical approach of content mining to analyze tweet contents and illustrate an application of the approach to the topic of physical activity [24].

Twitter data constitutes a rich source that can be used for capturing information about any topic imaginable. In this work, we use text mining to mine the Twitter health-related data. Text mining is the application of natural language processing techniques to derive relevant information [2]. Millions of tweets are generated each day on multifarious issues [15]. Twitter mining in large scale has been getting a lot of attention last few years. Lin and Ryaboy discussed the evolution of Twitter infrastructure and the development of capabilities for data mining on "big data" [13]. Pandarachalil et al. provided a scalable and distributed solution using Parallel python framework for Twitter sentiment analysis [15]. Large-scale Twitter Mining for drug-related adverse events was developed by Bian et al. [3].

In this paper, we use parallel and distributed technology Apache Kafka [11] to handle the large streaming twitter data. The data processing is conducted in parallel with data extraction by integration of Apache Kafka and Spark Streaming. Then we use Topic Modeling to infer semantic structure of the unstructured data (i.e Tweets). Topic Modeling is a text mining technique which automatically discovers the hidden themes from given documents. It is an unsupervised text analytic algorithm that is used for finding the group of words from the given document. We build the model using three different algorithms Latent Semantic Analysis (LSA) [7], Non-negative Matrix Factorization (NMF) [12], and Latent Dirichlet Allocation (LDA) [4] and infer the topic of tweets. To observe the model behavior, we test the model to infer new tweets. The implication of our work is to annotate unlabeled data using the model and find interesting correlation.

2 DATA COLLECTION

Tweet messages are retrieved from the Twitter source by utilizing the Twitter API and stored in Kafka topics. The Producer API is used to connect the source (i.e. Twitter) to any Kafka topic as a stream of records for a specific category. We fetch data from a source (Twitter), push it to a message queue, and consume it for further

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WISDOM@KDD'19, August 04–08, 2019, Anchorage, Alaska, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

analysis. Fig. 1 shows the overview of Twitter data collection using Kafka.

2.1 Apache Kafka

In order to handle the large streaming twitter data, we use parallel and distributed technology for big data framework. In this case, the output of the twitter crawling is queued in messaging system called Apache Kafka. This is a distributed streaming platform created and open sourced by LinkedIn in 2011 [11]. We write a Producer Client which fetches latest tweets continuously using Twitter API and push them to single node Kafka Broker. There is a Consumer that reads data from Kafka (Fig. 1).

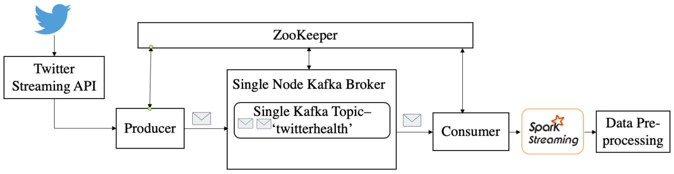


Figure 1: Twitter Data Collection.

2.2 Apache Zookeeper

Apache Zookeeper is a distributed, open-source configuration, synchronization service along with naming registry for distributed applications. Kafka uses Zookeeper to store metadata about the Kafka cluster, as well as consumer client details.

2.3 Data Extraction using Tweepy

The twitter data has been crawled using Tweepy which is a Python library for accessing the Twitter API. We use Twitter streaming API to extract 40k tweets (April 17-19, 2019). For the crawling, we focus on several keywords that are related to health. The keywords are processed in a non-case-sensitive way. We use filter to stream all tweets containing the word 'yoga', 'healthylife', 'healthydiet', 'diet','hiking', 'swimming', 'cycling', 'yogi', 'fatburn', 'weightloss', 'pilates', 'zumba', 'nutritiousfood', 'wellness', 'fitness', 'workout', 'vegetarian', 'vegan', 'lowcarb', 'glutenfree', 'calorieburn'.

The streaming API returns tweets, as well as several other types of messages (e.g. a tweet deletion notice, user update profile notice, etc), all in JSON format. We use Python libraries json for parsing the data, pandas for data manipulation.

2.4 Data Pre-processing

Data pre-processing is one of the key components in many text mining algorithms [2]. Data cleaning is crucial for generating a useful topic model. We have some prerequisites i.e. we download the stopwords from NLTK (Natural Language Toolkit) and spacy's en model for text pre-processing.

It is noticeable that the parsed full-text tweets have many emails, 'RT', newline and extra spaces that is quite distracting. We use Python Regular Expressions (re module) to get rid of them. Then we tokenize each text into a list of words, remove punctuation and unnecessary characters. We use Python Gensim package for further processing. Gensim's simple_preprocess() is used for tokenization and removing punctuation. We use Gensim's Phrases model to build bigrams. Certain parts of English speech, like conjunctions ("for", "or") or the word "the" are meaningless to a topic model. These terms are called stopwords and we remove them from the token list. We use spacy model for lemmatization to keep only noun, adjective, verb, adverb. Stemming words is another common NLP technique to reduce topically similar words to their root. For example, "connect", "connecting", "connected", "connection", "connections" all have similar meanings; stemming reduces those terms to "connect". The Porter stemming algorithm [16] is the most widely used method.

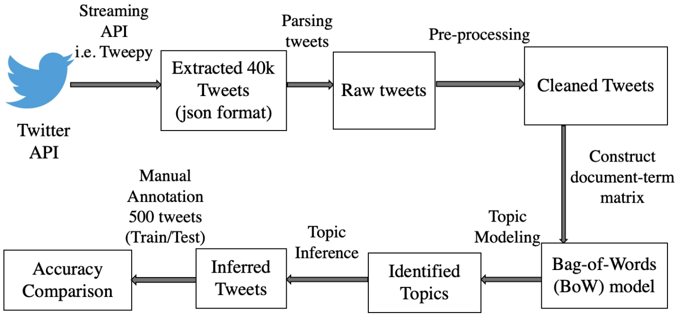


Figure 2: Methodology of correlation mining of Twitter health data.

3 METHODOLOGY

We use Twitter health-related data for this analysis. In subsections 3.1, 3.2, 3.3, and 3.4 elaborately present how we can infer the meaning of unstructured data. Subsection 3.5 shows how we do manual annotation for ground truth comparison. Fig. 2 shows the overall pipeline of correlation mining.

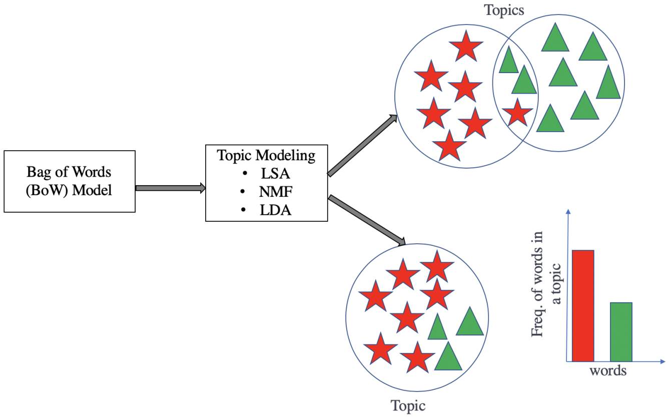
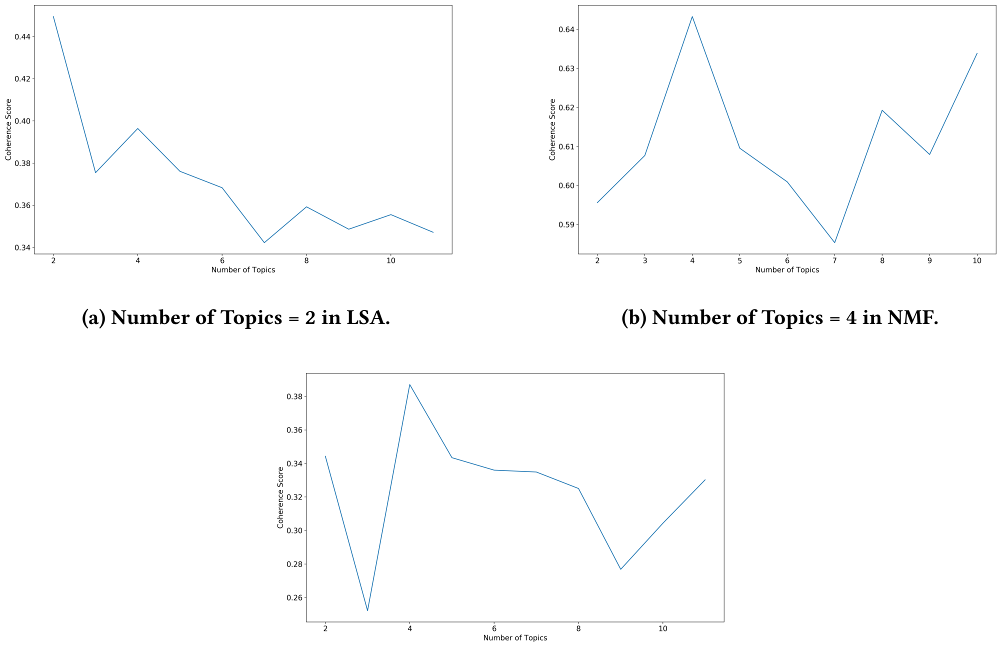


Figure 3: Topic Modeling using LSA, NMF, and LDA. After topic modeling we identify topic/topics (circles). Red pentagrams and green triangles represent group of co-occuring related words of corresponding topic.



(a) Number of Topics = 2 in LSA.

(b) Number of Topics = 4 in NMF.

(c) Number of Topics = 4 in LDA.

Figure 4: Optimal Number of Topics vs Coherence Score. Number of Topics (k) are selected based on the highest coherence score. Graphs are rendered in high resolution and can be zoomed in.

3.1 Construct document-term matrix

The result of the data cleaning stage is texts, a tokenized, stopped, stemmed and lemmatized list of words from a single tweet. To understand how frequently each term occurs within each tweet, we construct a document-term matrix using Gensim's Dictionary() function. Gensim's doc2bow() function converts dictionary into a bag-of-words. In the bag-of-words model, each tweet is represented by a vector in a m-dimensional coordinate space, where m is number of unique terms across all tweets. This set of terms is called the corpus vocabulary.

3.2 Topic Modeling

Topic modeling is a text mining technique which provides methods for identifying co-occurring keywords to summarize collections of textual information. This is used to analyze collections of documents, each of which is represented as a mixture of topics, where each topic is a probability distribution over words [1]. Applying these models to a document collection involves estimating the topic distributions and the weight each topic receives in each document. A number of algorithms exist for solving this problem. We use three unsupervised machine learning algorithms to explore the topics of the tweets: Latent Semantic Analysis (LSA) [7], Non-negative Matrix Factorization (NMF) [12], and Latent Dirichlet Allocation (LDA) [4]. Fig. 3 shows the general idea of topic modeling methodology. Each tweet is considered as a document. LSA, NMF, and LDA use Bag of Words (BoW) model, which results in a term-document matrix (occurrence of terms in a document). Rows represent terms (words) and columns represent documents (tweets). After completing topic modeling, we identify the groups of co-occurring words in tweets. These group co-occurring related words makes "topics".

3.2.1 *Latent Semantic Analysis (LSA).*

LSA (Latent Semantic Analysis) [7] is also known as LSI (Latent Semantic Index). It learns latent topics by performing a matrix decomposition on the document-term matrix using Singular Value Decomposition (SVD) [9]. After corpus creation in Subsection 3.1, we generate an LSA model using Gensim.

3.2.2 *Non-negative Matrix Factorization (NMF).*

Non-negative Matrix Factorization (NMF) [12] is a widely used tool for the analysis of high-dimensional data as it automatically extracts sparse and meaningful features from a set of non-negative data vectors. It is a matrix factorization method where we constrain the matrices to be non-negative.

We apply Term Weighting with term frequency-inverse document frequency (TF-IDF) [21] to improve the usefulness of the document-term matrix (created in Subsection 3.1) by giving more weight to the more "important" terms. In Scikit-learn, we can generate at TF-IDF weighted document-term matrix by using TfidfVectorizer. We import the NMF model class from sklearn.decomposition and fit the topic model to tweets.

3.2.3 *Latent Dirichlet Allocation (LDA).*

Latent Dirichlet Allocation (LDA) [4] is widely used for identifying the topics in a set of documents, building on Probabilistic Latent Semantic Analysis (PLSI) [10]. LDA considers each document as a collection of topics in a

Table 1: Topics and top-10 keywords of the corresponding topic

LSA		NMF				LDA			
Topic 1	*Topic 2*	*Topic 1*	*Topic 2*	*Topic 3*	*Topic 4*	*Topic 1*	*Topic 2*	*Topic 3*	*Topic 4*
Yoga	diet	Yoga	diet	vegan	fitness	diet	vegan	swimming	fitness
everi	vegan	job	beyonce	go	workout	workout	yoga	swim	amp
Life	fit	every_woman	new	eat	go	new	job	day	wellness
Job	day	cooks_goe	bitch	make	good	go	every_woman	much	health
Remember	new	therapy_remember	ciara_prayer	food	amp	day	cooks_goe	support	time
goe	like	life_juggl	day	day	day	beyonce	therapy_remember	really	great
Woman	Beyonce	everyone_birthday	eat	amp	yoga	get	life_juggle	try	look
Everyone	amp	boyfriend	go	shit	health	today	everyone_birthday	always	hiking
cook	eat	hot	fat	meat	gym	bitch	eat	relationship	make
therapy	workout	know	keto	vegetarian	today	gym	boyfriend	pool	love

certain proportion and each topic as a collection of keywords in a certain proportion. We provide LDA the optimal number of topics, it rearranges the topics' distribution within the documents and keywords' distribution within the topics to obtain a good composition of topic-keywords distribution.

We have corpus generated in Subsection 3.1 to train the LDA model. In addition to the corpus and dictionary, we provide the number of topics as well.

3.3 Optimal number of Topics

Topic modeling is an unsupervised learning, so the set of possible topics are unknown. To find out the optimal number of topic, we build many LSA, NMF, LDA models with different values of number of topics (k) and pick the one that gives the highest coherence score. Choosing a 'k' that marks the end of a rapid growth of topic coherence usually offers meaningful and interpretable topics.

We use Gensim's coherencemodel to calculate topic coherence for topic models (LSA and LDA). For NMF, we use a topic coherence measure called TC-W2V. This measure relies on the use of a word embedding model constructed from the corpus. So in this step, we use the Gensim implementation of Word2Vec [14] to build a Word2Vec model based on the collection of tweets.

We achieve the highest coherence score = 0.4495 when the number of topics is 2 for LSA, for NMF the highest coherence value is 0.6433 for K = 4, and for LDA we also get number of topics is 4 with the highest coherence score which is 0.3871 (see Fig. 4).

For our dataset, we picked k = 2, 4, and 4 with the highest coherence value for LSA, NMF, and LDA correspondingly (Fig. 4). Table 1 shows the topics and top-10 keywords of the corresponding topic. We get more informative and understandable topics using LDA model than LSA. LSA decomposed matrix is a highly dense matrix, so it is difficult to index individual dimension. LSA is unable to capture the multiple meanings of words. It offers lower accuracy than LDA.

In case of NMF, we observe same keywords are repeated in multiple topics. Keywords "go", "day" both are repeated in Topic 2, Topic 3, and Topic 4 (Table 1). In Table 1 keyword "yoga" has been found both in Topic 1 and Topic 4. We also notice that keyword "eat" is in Topic 2 and Topic 3 (Table 1). If the same keywords being repeated in multiple topics, it is probably a sign that the 'k' is large though we achieve the highest coherence score in NMF for k=4.

We use LDA model for our further analysis. Because LDA is good in identifying coherent topics where as NMF usually gives incoherent topics. However, in the average case NMF and LDA are similar but LDA is more consistent.

3.4 Topic Inference

After doing topic modeling using three different method LSA, NMF, and LDA, we use LDA for further analysis i.e. to observe the dominant topic, 2nd dominant topic and percentage of contribution of the topics in each tweet of training data. To observe the model behavior on new tweets those are not included in training set, we follow the same procedure to observe the dominant topic, 2nd dominant topic and percentage of contribution of the topics in each tweet on testing data. Table 2 shows some tweets and corresponding dominant topic, 2nd dominant topic and percentage of contribution of the topics in each tweet.

3.5 Manual Annotation

To calculate the accuracy of model in comparison with ground truth label, we selected top 500 tweets from train dataset (40k tweets). We extracted 500 new tweets (22 April, 2019) as a test dataset. We did manual annotation both for train and test data by choosing one topic among the 4 topics generated from LDA model (7th, 8th, 9th, and 10th columns of Table 1) for each tweet based on the intent of the tweet. Consider the following two tweets:

Tweet 1: *Learning some traditional yoga with my good friend.*

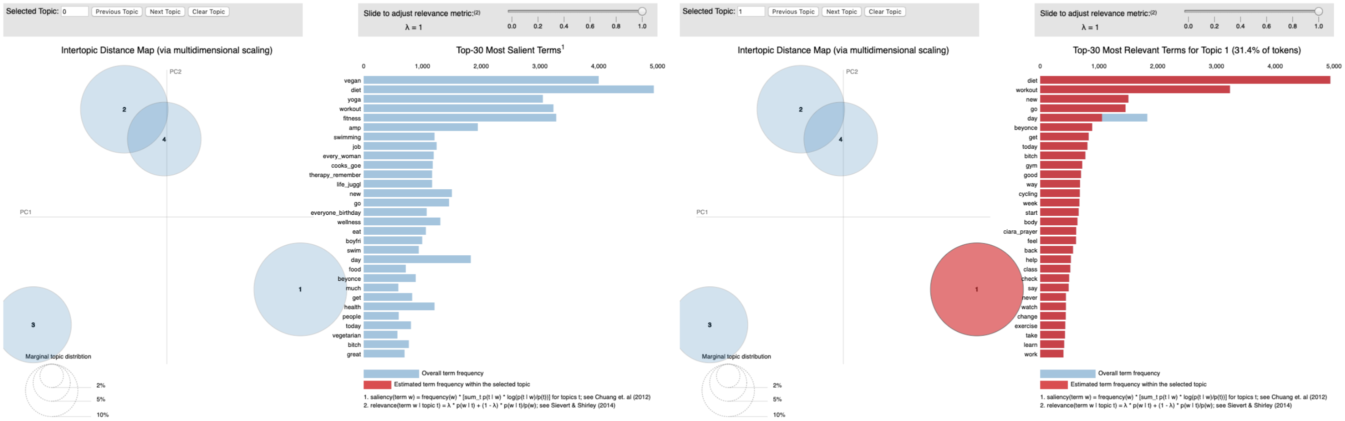
Tweet 2: *Why You Should #LiftWeights to Lose #BellyFat #Fitness #core #abs #diet #gym #bodybuilding #workout #yoga*

The intention of Tweet 1 is yoga activity (i.e. learning yoga). Tweet 2 is more about weight lifting to reduce belly fat. This tweet is related to workout. When we do manual annotation, we assign Topic 2 in Tweet 1, and Topic 1 in Tweet 2. It's not wise to assign Topic 2 for both tweets based on the keyword "yoga". During annotation, we focus on functionality of tweets.

4 RESULTS AND DISCUSSION

4.1 Visualization

We use LDAvis [22], a web-based interactive visualization of topics estimated using LDA. Gensim's pyLDAVis is the most commonly used visualization tool to visualize the information contained in

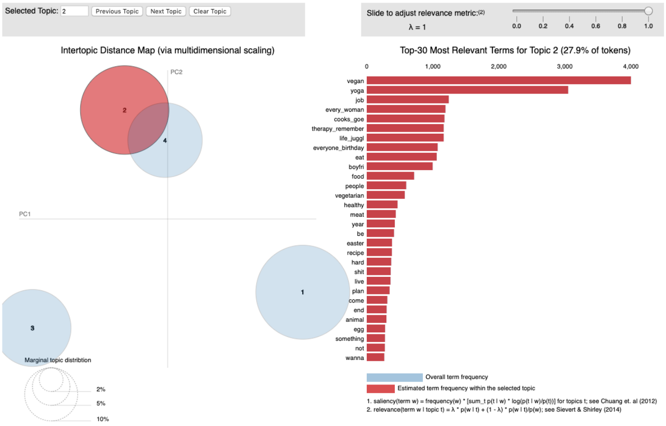


1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

(a) Bubbles in left hand side show overall topic distribution and sky blue bars in right hand side represent overall term frequencies. Best viewed in electronic format (zoomed in).

(b) Red bubble in left hand side represents the selected Topic which is Topic 1. Red bars in right hand side show estimated term frequencies of top-30 salient keywords that form the Topic 1. Best viewed in electronic format (zoomed in).

Figure 5: Visualization using pyLDAVis. Best viewed in electronic format (zoomed in).



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 6: Visualization using pyLDAVis. Red bubble in left hand side represents selected Topic which is Topic 2. Red bars in right hand side show estimated term frequencies of top-30 salient keywords that form the Topic 2. Best viewed in electronic format (zoomed in)

a topic model. In Fig. 5, each bubble on the left-hand side plot represents a topic. The larger the bubble, the more prevalent is that topic. A good topic model has fairly big, non-overlapping bubbles scattered throughout the chart instead of being clustered in one quadrant. A model with too many topics, is typically have many overlaps, small sized bubbles clustered in one region of the chart. In right hand side, the words represent the salient keywords.

If we move the cursor over one of the bubbles (Fig. 5b), the words and bars on the right-hand side have been updated and top-30 salient keywords that form the selected topic and their estimated term frequencies are shown.

We observe interesting hidden correlation in data. Fig. 6 has Topic 2 as selected topic. Topic 2 contains top-4 co-occurring keywords "vegan", "yoga", "job", "every_woman" having the highest term frequency. We can infer different things from the topic that "women usually practice yoga more than men", "women teach yoga and take it as a job", "Yogi follow vegan diet". We would say there are noticeable correlation in data i.e. ‘Yoga-Veganism’, ‘Women-Yoga’.

4.2 Topic Frequency Distribution

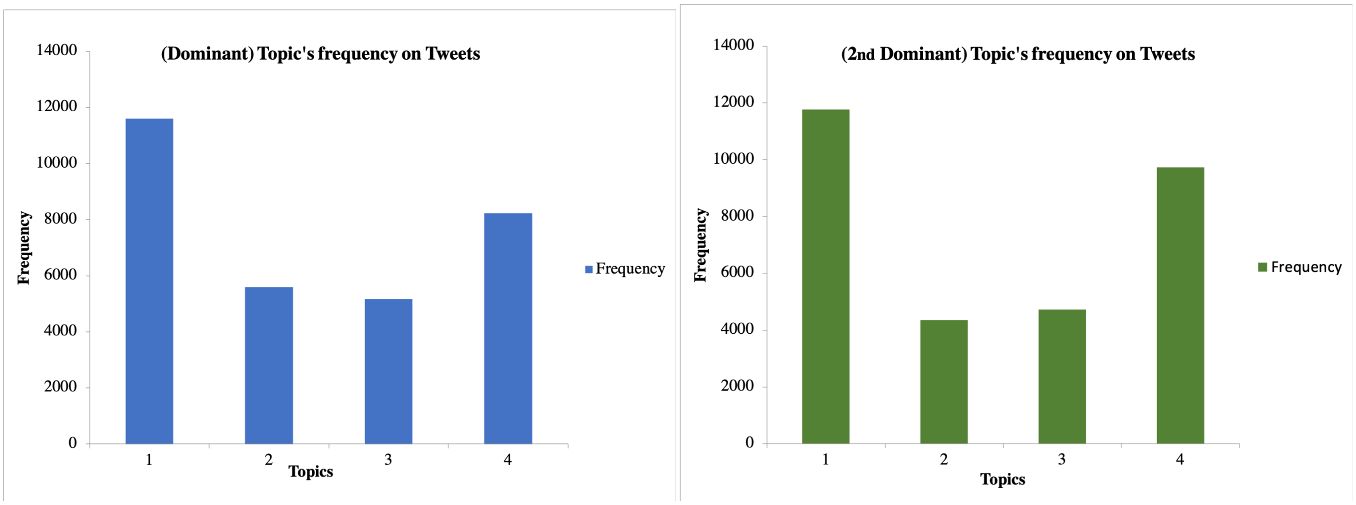
Each tweet is composed of multiple topics. But, typically only one of the topics is dominant. We extract the dominant and 2^{nd} dominant topic for each tweet and show the weight of the topic (percentage of contribution in each tweet) and the corresponding keywords.

We plot the frequency of each topic’s distribution on tweets in histogram. Fig. 7a shows the dominant topics’ frequency and Fig. 7b shows the 2^{nd} dominant topics’ frequency on tweets. From Fig. 7 we observe that Topic 1 became either the dominant topic or the 2^{nd} dominant topic for most of the tweets. 7^{th} column of Table 1 shows the corresponding top-10 keywords of Topic 1.

4.3 Comparison with Ground Truth

To compare with ground truth, we gradually increased the size of dataset 100, 200, 300, 400, and 500 tweets from train data and test data (new tweets) and did manual annotation both for train/test data based on functionality of tweets (described in Subsection 3.5).

For accuracy calculation, we consider the dominant topic only. We achieved 66% train accuracy and 51% test accuracy when the size of dataset is 500 (Fig. 8). We did baseline implementation with random inference by running multiple times with different seeds and took the average accuracy. For dataset 500, the accuracy converged towards 25% which is reasonable as we have 4 topics.



(a) Dominant topic.

(b) 2^{nd} dominant topic.

Figure 7: Frequency of each topic's distribution on tweets.

Table 2: The Dominant & 2^{nd} Dominant Topic of a Tweet and corresponding Topic Contribution on that specific Tweet.

Dataset	Tweets	Dominant Topic	Contribution (%)	2^{nd} Dominant Topic	Contribution (%)
Train	Revoking my vegetarian status till further notice. There's something I wanna do and I can't afford the supplements that come with being veggie.	2	61	1	18
Test	I would like to take time to wish "ALL" a very happy #EarthDay! #yoga #meditation	2	33	4	32
Test	This morning I packed myself a salad. Went to yoga during lunch. And then ate my salad with water in hand. I'm feeling so healthy I don't know what to even do with myself. Like maybe I should eat a bag of chips or something.	2	43	3	23
Test	My extra sweet halfcaf double vegan soy chai pumpkin latte was 2 degrees hotter than it should have been and the foam wasn't very foamy. And they spelled my name Jimothy, "Jim" on the cup. it's a living hell here.	3	37	2	33

4.4 Observation and Future Work

In Table 2, we show some observations. For the tweets in 1^{st} and 2^{nd} row (Table 2), we observed understandable topic. We also noticed misleading topic and unrelated topic for few tweets (3^{rd} and 4^{th} row of Table 2).

In the 1^{st} row of Table 2, we show a tweet from train data and we got Topic 2 as a dominant topic which has 61% of contribution in this tweet. Topic 1 is 2^{nd} dominant topic and 18% contribution here.

2^{nd} row of Table 2 shows a tweet from test set. We found Topic 2 as a dominant topic with 33% of contribution and Topic 4 as 2^{nd} dominant topic with 32% contribution in this tweet.

In the 3^{rd} (Table 2), we have a tweet from test data and we got Topic 2 as a dominant topic which has 43% of contribution in this tweet. Topic 3 is 2^{nd} dominant with 23% contribution which is misleading topic. The model misinterprets the words 'water in hand' and infers topic which has keywords "swimming, swim, pool". But the model should infer more reasonable topic (Topic 1 which has keywords "diet, workout") here.

We got Topic 2 as dominant topic for the tweet in 4^{th} row (Table 2) which is unrelated topic for this tweet and most relevant topic of this tweet (Topic 2) as 2^{nd} dominant topic. We think during accuracy comparison with ground truth 2^{nd} dominant topic might be considered.

In future, we will extract more tweets and train the model and observe the model behavior on test data. As we found misleading and unrelated topic in test cases, it is important to understand the reasons behind the predictions. We will incorporate Local Interpretable model-agnostic Explanation (LIME) [20] method for the explanation of model predictions. We will also do predictive causality analysis on tweets.

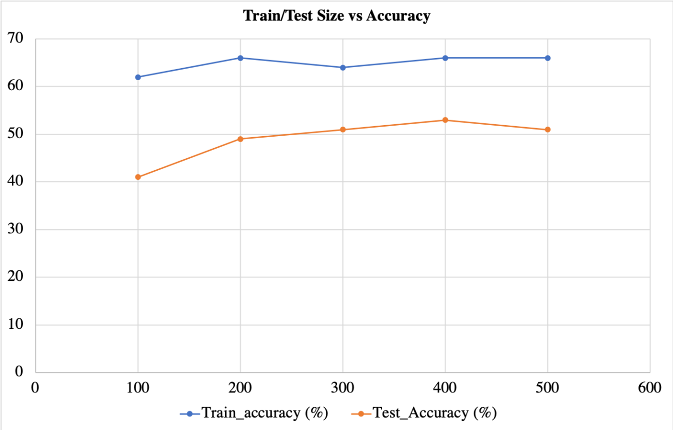


Figure 8: Percentage of Accuracy (y-axis) vs Size of Dataset (x-axis). Size of Dataset = 100, 200, 300, 400, and 500 tweets. Blue line shows the accuracy of Train data and Orange line represents Test accuracy. Best viewed in electronic format (zoomed in).

5 CONCLUSIONS

It is challenging to analyze social media data for different application purpose. In this work, we explored Twitter health-related data, inferred topic using topic modeling (i.e. LSA, NMF, LDA), observed model behavior on new tweets, compared train/test accuracy with ground truth, employed different visualizations after information integration and discovered interesting correlation (Yoga-Veganism) in data. In future, we will incorporate Local Interpretable model-agnostic Explanation (LIME) method to understand model interpretability.

REFERENCES

[1] Rubayyi Alghamdi and Khalid Alfalqi. 2015. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)* 6, 1 (2015).

[2] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919* (2017).

[3] Jiang Bian, Umit Topaloglu, and Fan Yu. 2012. Towards large-scale twitter mining for drug-related adverse events. In *Proceedings of the 2012 international workshop on Smart health and wellbeing*. ACM, 25–32.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3 (2003), 993–1022.

[5] Nathan K Cobb and Amanda L Graham. 2012. Health behavior interventions in the age of facebook. *American journal of preventive medicine* 43, 5 (2012), 571–572.

[6] Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. In *Seventh international AAAI conference on weblogs and social media*.

[7] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41 (1990), 391–407.

[8] Johannes C Eichstaedt, Robert J Smith, Raina M Merchant, Lyle H Ungar, Patrick Crutchley, Daniel Preoţiuc-Pietro, David A Asch, and H Andrew Schwartz. 2018. Facebook language predicts depression in medical records. *Proceedings of the National Academy of Sciences* 115, 44 (2018), 11203–11208.

[9] Gene H Golub and Christian Reinsch. 1971. Singular value decomposition and least squares solutions. In *Linear Algebra*. Springer, 134–151.

[10] Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. 289–296.

[11] Jay Kreps, Neha Narkhede, Jun Rao, et al. 2011. Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*. 1–7.

[12] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. (2001).

[13] Jimmy Lin and Dmitriy Ryaboy. 2013. Scaling big data mining infrastructure: the twitter experience. *Acm SIGKDD Explorations Newsletter* 14, 2 (2013), 6–19.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. (2013).

[15] Rafeeque Pandarachalil, Selvaraju Sendhilkumar, and GS Mahalakshmi. 2015. Twitter sentiment analysis for large-scale data: an unsupervised approach. *Cognitive computation* 7, 2 (2015), 254–262.

[16] Martin F Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.

[17] Kyle W Prier, Matthew S Smith, Christophe Giraud-Carrier, and Carl L Hanson. 2011. Identifying health-related topics on twitter. In *International conference on social computing, behavioral-cultural modeling, and prediction*. Springer, 18–25.

[18] Víctor M Prieto, Sergio Matos, Manuel Alvarez, Fidel Cacheda, and José Luís Oliveira. 2014. Twitter: a good place to detect health conditions. *PloS one* 9, 1 (2014), e86191.

[19] Andrew G Reece, Andrew J Reagan, Katharina LM Lix, Peter Sheridan Dodds, Christopher M Danforth, and Ellen J Langer. 2017. Forecasting the onset and course of mental illness with Twitter data. *Scientific reports* 7, 1 (2017), 13006.

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[21] Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval. (1986).

[22] Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*. 63–70.

[23] Youngseo Son, Anneke Buffone, Joe Raso, Allegra Larche, Anthony Janocko, Kevin Zembroski, H Andrew Schwartz, and Lyle Ungar. 2017. Recognizing counterfactual thinking in social media texts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 654–658.

[24] Sunmoo Yoon, Noémie Elhadad, and Suzanne Bakken. 2013. A practical approach for content mining of tweets. *American journal of preventive medicine* 45, 1 (2013), 122–129.