Spring 2018

Subtopics in Yelp Reviews

Riya Suchdev
San Jose State University

Subtopics in Yelp Reviews

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Riya Suchdev

May 2018

The Designated Project Committee Approves the Project Titled

Subtopics in Yelp Reviews

by

Riya Suchdev

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

May 2018

Dr. Katerina Potika	Department of Computer Science
Dr. Sami Khuri	Department of Computer Science
Rahul Ravindran	Software Engineer, Infrastructure at Yelp

ABSTRACT

Subtopics in Yelp Reviews

by Riya Suchdev

Yelp is a review platform that connects people to local businesses. It is a very popular platform that helps customers decide which business to choose. It relies on crowd sourced plain text reviews. From the business's description some facts can be determined, such as category and location. However, more detailed description can be extracted from the reviews. Discovering latent topics and subtopics in Yelp reviews, can help summarize the reviews to gain knowledge. For example, we can deduce that reviews related to the Restaurant category tend to emphasize on service, food, order etc. Additionally, one can deduce positive or negative feedback on each topic and subtopic. In this project, we study the problem of content topic discovery using probabilistic and other models in a Yelp dataset. Various experiments were performed to extract word features, by trying to keep the initial context and sentence structure with the use techniques such as Document to bag of words, Word Embedding, Parts of Speech (POS) tagging and Term Frequency-Inverse Document Frequency (TF-IDF). In our approach, we discover topics in the Yelp corpus with the use of Machine Learning techniques. Specifically, we use the Latent Dirichlet Allocation (LDA), the Latent Semantic Analysis (LSA) and the K-Means technique. These unsupervised learning techniques divide the corpus into latent topics that summarize the review text and highlights the insight of it. The methods are compared using the Coherence Model and the resultant LDA model is visualized using pyLDAvis. Finally, by comparing our techniques, we conclude that the K-Means using Word Embeddings on particular Parts of Speech tagged words gives best results, but is time consuming. On the contrary, LDA applied on cleaned corpus containing POS tagged words with TF-IDF

is much faster albeit topics report loss of context in comparison to K-Means.

ACKNOWLEDGMENTS

This project work could not have been possible without the help of friends, family members and the instructors who have supported me and guided me throughout the project work. I would like to specifically thank my project advisor, Dr. Katerina Potika for guiding me through the course of project work. This project could not have been possible without her continuous efforts and her wisdom. Also, It was her pure perseverance that pushed me to perform better during the project that significantly contributed to wards its completion on schedule. I would also like to thank the members of the Committee , Dr. Sami Khuri and Mr. Rahul Ravindran for their continuous guidance and support. It gives you great encouragement when you know that there are people whom you can reach out to whenever you get stuck at something.

Finally, I would like to thank my parents Mr. Prem Suchdev and Mrs. Jaya Suchdev , family members , and friends for their perennial encouragement and emotional support.

TABLE OF CONTENTS

CHAPTER

1 Introduction 1

2 Motivation and Problem Definition 3

3 Notation and Techniques 7

3.1 Term Frequency/Inverse Document Frequency 7

3.2 Latent Dirichlet Allocation 8

3.3 Coherence score 9

4 Related Works 10

5 Framework/Algorithms/Techniques 12

5.1 Data 12

5.2 Data Cleaning 15

5.2.1 Misspelling or alternately spelled words 17

5.2.2 Punctuation and Numerical Removal 17

5.2.3 Language variability 17

5.2.4 POS Tags 18

5.3 Word to Corpus 18

5.3.1 Doc2Bow 18

5.3.2 Word Embedding using Word2vec 18

5.4 Machine Learning Models 20

5.4.1 Latent semantic analysis 21

5.4.2 Latent Dirichlet Allocation 21

5.4.3 OnlineLDA 22
5.4.4 Hierarchical Dirichlet Processing 23
5.4.5 K-Means 23
5.5 Libraries used for Data cleaning 24
5.5.1 PyEnchant 24
5.5.2 SpaCy 24
5.5.3 Gensim 25
5.6 Visualization 25
6 Experiments and Results 27
6.1 Data Cleaning 27
6.2 Methods in Experiments 28
6.3 Whole Yelp 29
6.3.1 Visualization 31
6.4 Restaurants 31
6.5 Hotels 37
6.6 Panda Express 38
6.7 Restaurants in Las Vegas 39
6.8 Other Observations 39
6.8.1 Reference based topic context 39
6.8.2 Positive and Negative sentiments in Subtopic 40
6.9 Model Comparison 40
6.9.1 Coherence score 40
6.9.2 Time to implement 41

7 Conclusion and Future Work 43

7.1 Conclusion 43

7.2 Future Work 44

LIST OF REFERENCES 45

LIST OF TABLES

1	Topics using K-Means	30
2	Topics using K-Means	30
3	5 largest Topics in LDA and top 10 words in each	31
4	5 largest Topics in HDP and top 10 words in each	32
5	5 largest Topics in LSI and top 10 words in each	33
6	5 largest Topics in OnlineLDA and top 10 words in each	33
7	5 largest Topics in LDA and top 10 words in each for Restaurants	37
8	Topics using K-Means on Restaurants	37
9	Topics using K-Means on Hotels	38
10	Topics using K-Means on Panda Express	38
11	Topics using K-Means on Las Vegas Restaurants	39
12	Positive and Negative Service Topics	40
13	Example of Service in Categories context	40
14	Time taken for 20000 reviews, K=50	42
15	Time taken for 2000 reviews, K=5	42

LIST OF FIGURES

1	Reviews, bag of words, parts of speech matching, name entity tags	3
2	Topic Modeling	4
3	Topic Modeling Flow	5
4	LDA Model	8
5	LDA parameters	9
6	Generative Model LDA	9
7	Example Review	13
8	Example Business JSON	13
9	Categories Distribution.	14
10	Number of Reviews Per City.	15
11	Number of Registered Businesses Per City.	16
12	Workflow	17
13	Bag of Words	19
14	CBOW and Skip-Gram Methods	19
15	LDA Topic Model	34
16	LDA Topic Model for Cluster 11	34
17	LDA Topic Model for Cluster 8	35
18	Topic Modeling	35
19	LDA topics containing word "sushi"	36
20	LDA topics containing word "service"	36
21	Coherence score for LSI, HDP, LDA and OnlineLDA	41

CHAPTER 1

Introduction

Over the past couple of years, social media has gained a lot of importance in our society. From major product launches by multinational companies to major announcements, social media is being used to give out and receive information. People use the information on these social media websites and applications to decide where to eat, when and where to go for a vacation etc. Text data is available everywhere and can be used to get insight about the current market and products available. It can also be used by business owners to validate, improve and expand on their current products and its functionalities. Natural Language Processing deals with extracting meaningful information from text.

Yelp is one of the applications used by people to decide which restaurant to eat at. Yelp has been impactful in the way customers gain information about products and services of a place. It has a huge impact on customer inflow for small and large restaurants alike. In some studies, it was observed that a change in single star rating can improve independent restaurant revenue between 5-9 %. [1] Restaurant owners also can use the application to receive feedback about how their restaurant is viewed. But due to the large number of reviews on Yelp, it is possible to miss out on an aspect of dining experience of the customers. This report focuses on the methodology to create a machine learning (ML) model that will take reviews for each restaurant and produce the most common topics that customers talk about in their review for the restaurant.

Subtopics will be a list of descriptive labels that summarize the content of the

reviews. In this report, we go through few of the ML techniques to leverage information from the unstructured text data contained in Yelp. Since the data are human generated, we have to assume that there will be language variability, misspellings and various other types of noise. Hence, we will discuss methods to efficiently remove noise from the text data and improve data quality given to the ML models. We use methods like stop words removal, stemming, lemmatization, term frequency-inverse document frequency to generate a clean corpus of words and feed the models to convert them into vectors. These vectors are then fed to Topic Modeling techniques like Latent Semantic Analysis (LSA/LSI), Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Process (HDP) and K-means. These unsupervised models divide the corpus into a K number of topics defined by the user. The value of K should be optimal so that there is no over-fitting or under fitting of words per topic. Various methods such as elbow method (for K-Means) and coherence score are used to find the optimal values of K.

The results are judged on the basis of how well the corpus is divided into topics. As there is no absolute metric to judge how well the topic is divided, human understanding is used to judge whether the model has grasped the topics correctly.

As a possible extension of this work, the techniques described in the report can be extended to various types of text analysis systems.

CHAPTER 2

Motivation and Problem Definition

Each document contains various topics in certain proportions. This structure is easy for humans to interpret as we have prior knowledge of words, their context and topics these words belong to. For example, in Figure 1, we can see that the document is made up of n number of topics. We can determine this knowledge by internally tagging each word and clustering words topics.

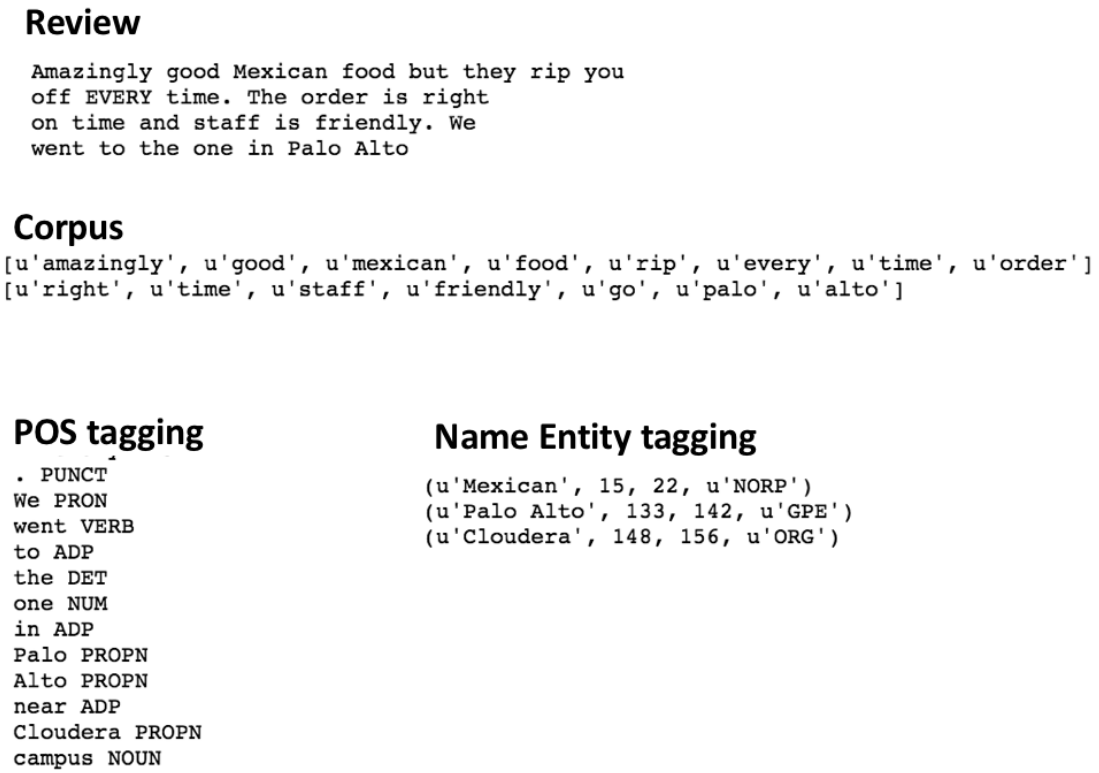

Figure 1: Reviews, bag of words, parts of speech matching, name entity tags

Inversely, various words belong to certain topics and if words of a topic appear in a document, we can determine that a part of document or the whole docu-

ment(depending on the number of words belonging to a topic vs the overall number of words) is made up that particular topic. [2]

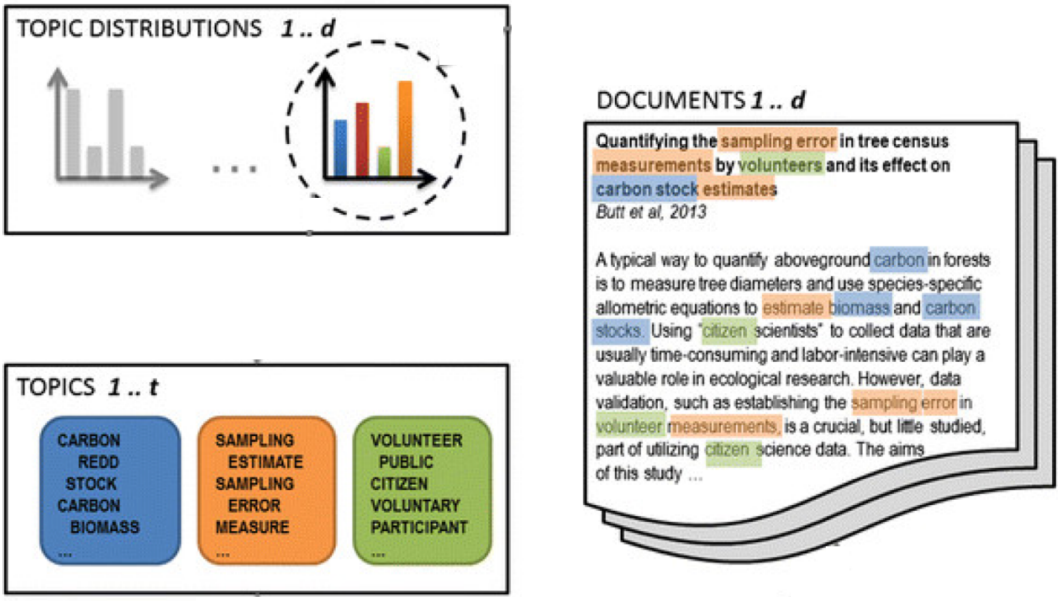

Figure 2: Topic Modeling

The web has huge dumps of unstructured text data that can be used to gain insights about various topics. Yelp has become a trusted source of information about local business. It has reviews on businesses ranging from restaurants to banks. Good reviews and ratings on Yelp can help attract more customers which is very helpful for small businesses and hence Yelp has approximately 86000 active small business accounts. [3] Reviews help businesses determine what keeps their customers happy and to improve on things they get bad reviews for, thus maintaining or potentially improving their service and in-turn their customer base. Having a ML model that describes the reviews in a few words would would prove helpful for the owners to keep a track of the customer's views towards their business.

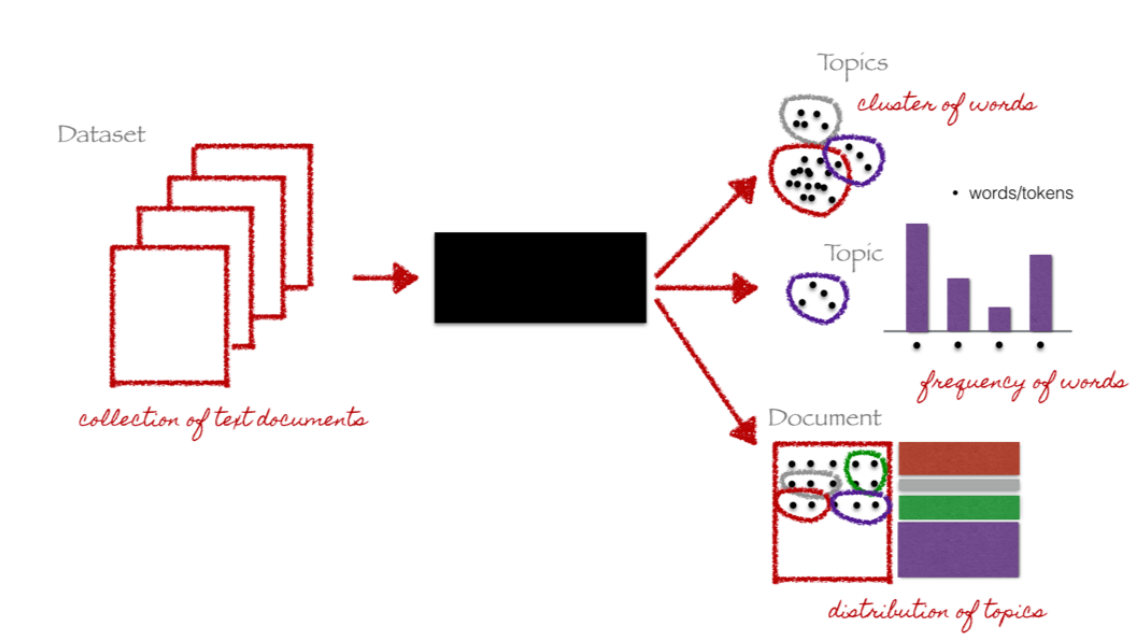

Figure 3: Topic Modeling Flow

There are around 148 million reviews on Yelp and has a presence in more than 30 countries around the world. [3] Reading through all reviews may be very time consuming in many cases. Each review may range from 100 to 1000 words and having more than 1000 to these reviews can be overwhelming and time consuming. We can leverage the power of various ML techniques to get the subtopics in a matter of minuets.

The goal is to analyze this large volume of unstructured, unlabeled text into "topics" that summarize these reviews. Words with similar meanings and contextual information are clustered into topics. At the end of this thesis project, we want to summarize reviews for a business and present it to the owners. We can also determine common topics relevant to customers in a certain locality(town, city), related to a particular type of business or which type of businesses are reviewed the most in a locality. Analysis of reviews can also be used to determine some interesting insights

about customer behavior and preferences in a locality where a business owner wants to start their business as they maximize their profit. Figure 3 gives the flow of the model [4].

CHAPTER 3

Notation and Techniques

Various notations and techniques used in the work are described in this chapter.

3.1 Term Frequency/Inverse Document Frequency

Tf-idf is a popular concept in information retrieval and is called term frequency-inverse document frequency. It is a statistic representing how important a word is to a document. Breaking down the terms, term frequency corresponds to the frequency of a word in a document. Therefore, if we find a work appearing 5 times in a document, its Term Frequency (TF) would be 5. It is natural for documents to contain a lot of common words. This includes stop words such as the, a, an etc. Considering only the TF, we would find that the stop words would have the highest weight. However, this should not be the case, here comes inverse document frequency.

Inverse Document Frequency (IDF) penalizes words that appear quite often and promotes words that appear far less common. This is because the rarer the word is, the more is the chance that it contains more information. Therefore, TF-IDF is a combination of both of these, it emphasizes term frequency but penalizes on the word appearing a lot in the collection. IDF is calculated as the logarithm of the total number of documents divided by the documents containing the word. So, if the documents containing a certain word is zero, its IDF value would be infinity and hence we add 1 to the denominator to avoid the infinity value.

TF-IDF thus becomes the TF multiplied by the inverse document frequency. TF-IDF is extensively used in search and text mining. Since, we can get a TF-IDF

value for each word, it is possible to represent each document in term of TF-IDF vector. Thus, if we were to use a distance formula such as cosine distance or some other form (euclidean distance), it is possible to find if one document is similar to other. This is the basis for most search engines too, where a search term is converted to a vector and TF-IDF values are compared on all the documents available where one or more of the words appear.

3.2 Latent Dirichlet Allocation

The LDA model has the structure [5] as in Figure 4.

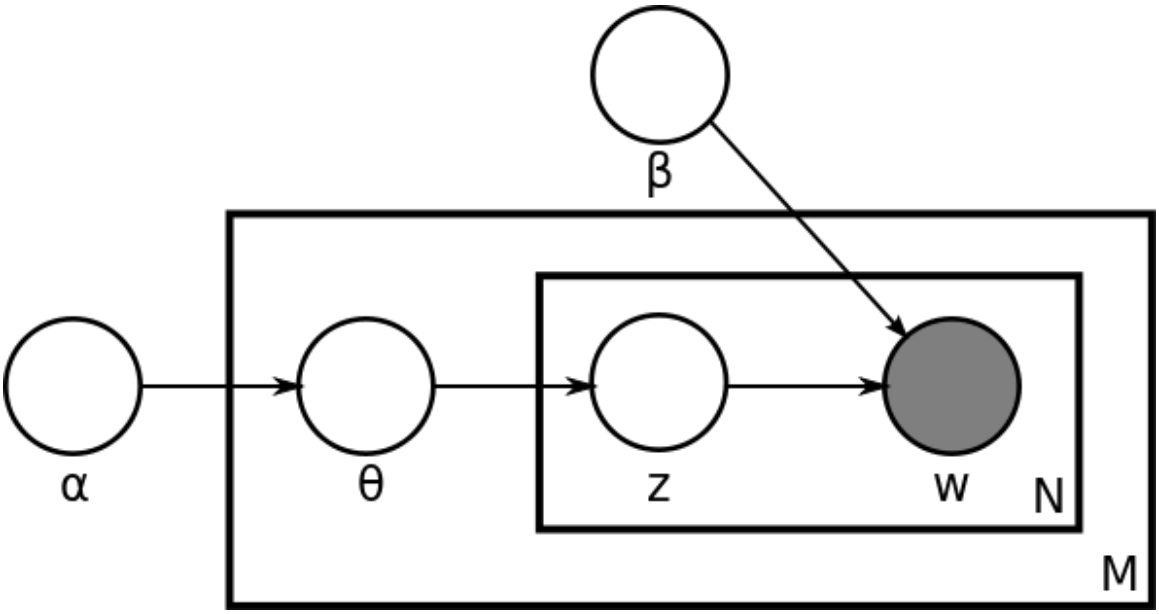

Figure 4: LDA Model

Its generative process is as follows: It is assumed that every document is made up of K topics. These topics are assumed to be latent. The topics are assumed to be made up of all words existing in the text corpus. LDA then assumes the following generative process for a corpus D consisting of M documents each of length N(i)

α is the parameter of the Dirichlet prior on the per-document topic distributions,
β is the parameter of the Dirichlet prior on the per-topic word distribution,
θ_m is the topic distribution for document m,
φ_k is the word distribution for topic k,
z_{mn} is the topic for the n-th word in document m, and
w_{mn} is the specific word.

Figure 5: LDA parameters

1. Choose $\theta_i \sim \mathrm{Dir}(\alpha)$, where $i \in \{1, \ldots, M\}$ and $\mathrm{Dir}(\alpha)$ is a Dirichlet distribution with a symmetric parameter α which typically is sparse ($\alpha < 1$)
2. Choose $\varphi_k \sim \mathrm{Dir}(\beta)$, where $k \in \{1, \ldots, K\}$ and β typically is sparse
3. For each of the word positions i, j, where $i \in \{1, \ldots, M\}$, and $j \in \{1, \ldots, N_i\}$
   (a) Choose a topic $z_{i,j} \sim \mathrm{Multinomial}(\theta_i)$.
   (b) Choose a word $w_{i,j} \sim \mathrm{Multinomial}(\varphi_{z_{i,j}})$.

Figure 6: Generative Model LDA

3.3 Coherence score

Coherence score of a model is calculated based on the coherence or interpretability of the topics as compared to the already existing models. Coherence model has models trained on WordNet, Wikipedia and the Google search engine [6]. Corpus of the data being used is run for the above models and topics are generated. These topics are then compared with the topics generated by our topic model. Then a significance score for the topic is computed using various types of dissimilarities and similarities to these three prototype models. The intrinsic measure - UMass and the extrinsic measure UCI are generally used to compare the topics generated by these 2 models. Usually the pairwise frequencies of top k elements in one model with respect to the other model is summed to get a coherence score. [7]

CHAPTER 4

Related Works

In their work, J. Huang, et. al. [8] uses LDA, an unsupervised learning algorithm that detects latent topics in a corpus and describes each topic with the probability of words occurring in topic. It assumes that each document is made up of these topics and clusters the words into these K topics. Using this method, the authors were able to find out various topics discussed in Yelp reviews. After careful analysis and evaluation of models, they concluded that K = 50 topics gives the optimal result. They used human intuition as a metric to judge the correctness of the model. Methods like Latent Semantic Analysis are also mentioned in paper by S. Deerwester, et. al. [9] are used to describe the topics. The results for LSI model are seen to be a bit worse than that of LDA as will be seen in the Experiments section. Words such as "lunch","service","breakfast" are the most common topics.

Another related paper by W. Dai, et.al. uses a traditional LDA [10] with review ratings along with text to extract latent topics from the corpus. They also divided the dataset into restaurants that have 1-star rating and 5-star rating and tried to see what was the difference in between restaurants with high star rating and low star rating. They discovered that in topics for 1-star restaurants, there were many negative words like "not", "slow", "longer" etc. appear more than positive words. Many other recent studies use unsupervised learning to improve rating prediction and recommendation accuracy by using groups of customers.

One of the new techniques employed in the paper by D. Hazarika, et. al. [11], use of word embeddings to cluster words into topics. The paper is submitted for

the NAACL conference in 2018. The paper describes the use of Word2Vec word embedding with K-Means to clusters word vectors, maintaining the word context and semantics of the review.

CHAPTER 5

Framework/Algorithms/Techniques

5.1 Data

The primary source of data is the Yelp Dataset provided on the Yelp website [12]. It consists of a subset of reviews of a set of restaurants from all across the world. The dataset itself is divided into various sections and it has JSON files available for each section. The sections consist of:

- reviews
- business
- user
- check-in

The review.json, as shown in Figure 7, consists of various fields and ones relevant to us include business-ids, review text, rating(rating of the review for the business). From the business.json file as shown in Figure 8, we use fields of business-id(to map to business ids in review.json), business names, categories, city, star-rating(overall star rating of the business).

There are about 5 million reviews in all for about 10000 different businesses across 1094 cities in US and around the world.

Figure 9 shows the top 30 categories and the distribution of number of reviews across the dataset. Categories

```
{u'business_id': u'uYHaNptLzDLoV_JZ_MuzUA',
 u'cool': 1,
 u'date': u'2016-11-08',
 u'funny': 1,
 u'review_id': u'JQJvnM3p-3eML05eKcTgiw',
 u'stars': 4,
 u'text': u"A hotel that has all the basics that you'd need - no additional fuss. For the price we paid, I'd say this
place is actually pretty good value. The location is super central and it's easy to get to everywhere from here! It's
also right next to the airport shuttle bus and train station, so transport is also easy. \n\nThe insides is a bit of
a maze depending on where your room is located - you may have to go down to go up to go back down again to get to you
r room! But no bother, it's good for walking off the chips and sobering from your drinks.",
 u'useful': 1,
 u'user_id': u'tgV6tsYQ66DZ3LQKvtC6cw'}
```


Figure 7: Example Review


```
{u'address': u'691 Richmond Rd',
 u'attributes': {u'BikeParking': True,
  u'BusinessParking': {u'garage': False,
   u'lot': True,
   u'street': False,
   u'valet': False,
   u'validated': False},
  u'RestaurantsPriceRange2': 2,
  u'WheelchairAccessible': True},
 u'business_id': u'YDf95gJZaq05wvo7hTQbbQ',
 u'categories': [u'Shopping', u'Shopping Centers'],
 u'city': u'Richmond Heights',
 u'hours': {u'Friday': u'10:00-21:00',
  u'Monday': u'10:00-21:00',
  u'Saturday': u'10:00-21:00',
  u'Sunday': u'11:00-18:00',
  u'Thursday': u'10:00-21:00',
  u'Tuesday': u'10:00-21:00',
  u'Wednesday': u'10:00-21:00'},
 u'is_open': 1,
 u'latitude': 41.5417162,
 u'longitude': -81.4931165,
 u'name': u'Richmond Town Square',
 u'neighborhood': u'',
 u'postal_code': u'44143',
 u'review_count': 17,
 u'stars': 2.0,
 u'state': u'OH'}
```


Figure 8: Example Business JSON

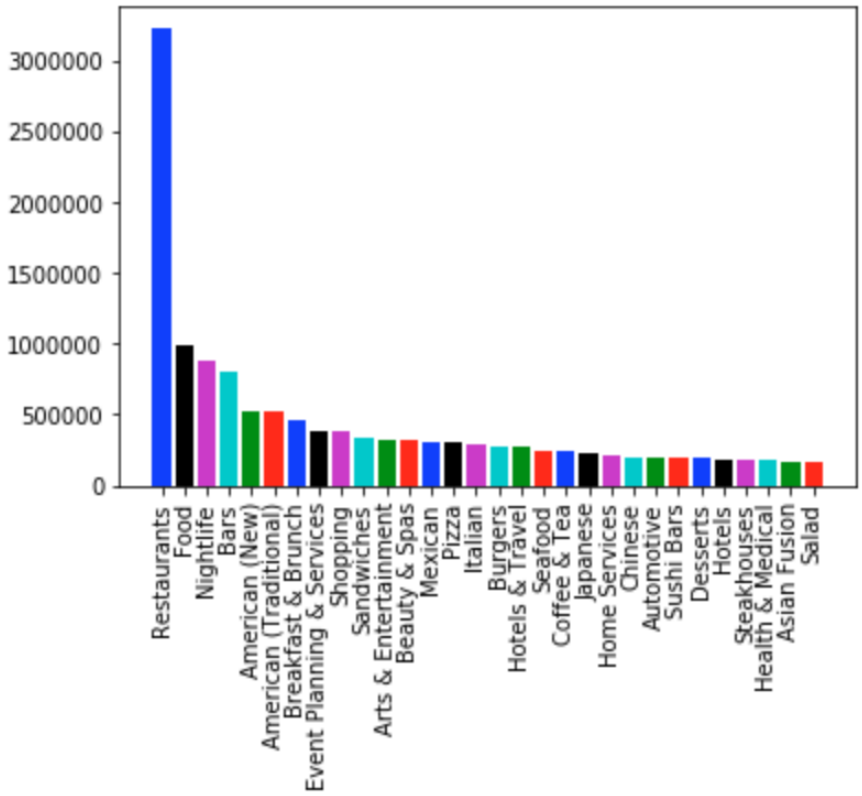

Figure 9: Categories Distribution.

Figure 10 shows the top 30 cities based on number of reviews per city. As seen, Las Vegas is the city that has most active reviewers.

Figure 11 shows the top 30 cities based on number of registered businesses per city. As seen, Las Vegas has the highest number of businesses registered.

The reviews file was about 3.7GB which was difficult to load into memory at once. Hence, the files were read one at a time and loaded to a dataframe.

From the above example, data is cleaned to create a Corpus. Figure 12 shows the workflow. Corpus is defined as the collection of words in each document. Corpus does

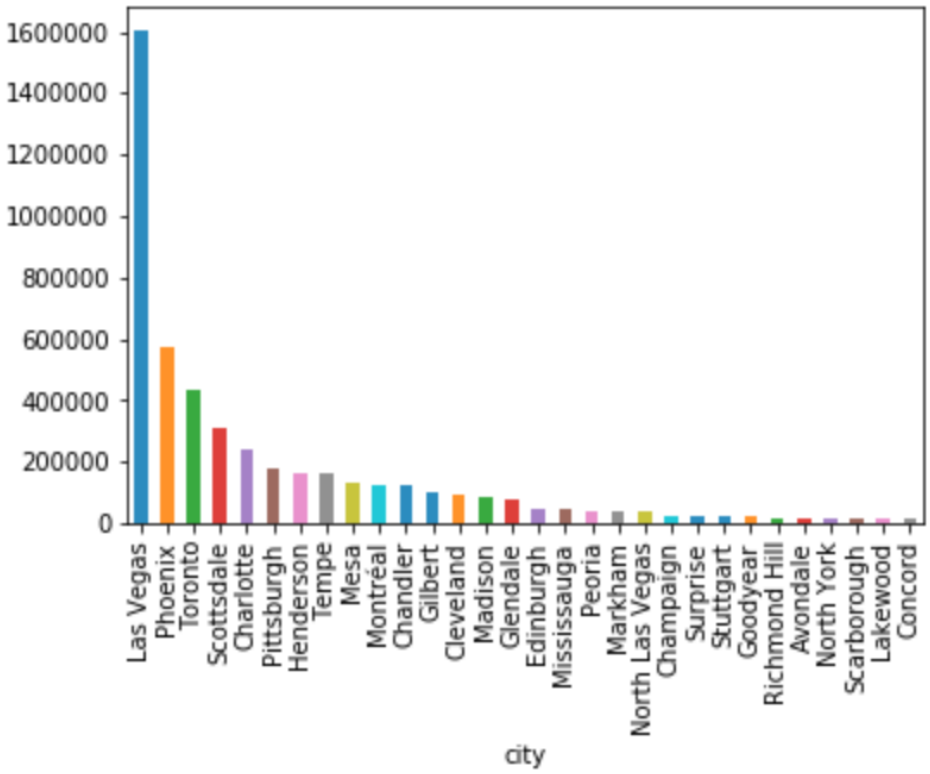

Figure 10: Number of Reviews Per City.

not store the words itself but the token ids for each word along with word frequency in the document. Word frequency can simply be the count of word per document or TF-IDF value assigned by us.

Dictionary of a corpus is the token id to word mapping done for the whole text data. This is then fed to data cleaning module, where the reviews are stripped down to their bare essentials to make the ML models work.

5.2 Data Cleaning

We first start by getting useful data and patterns from the bulk of text. It is important that we give the models relevant data as we are using unsupervised learning and bad data leads to bad results.

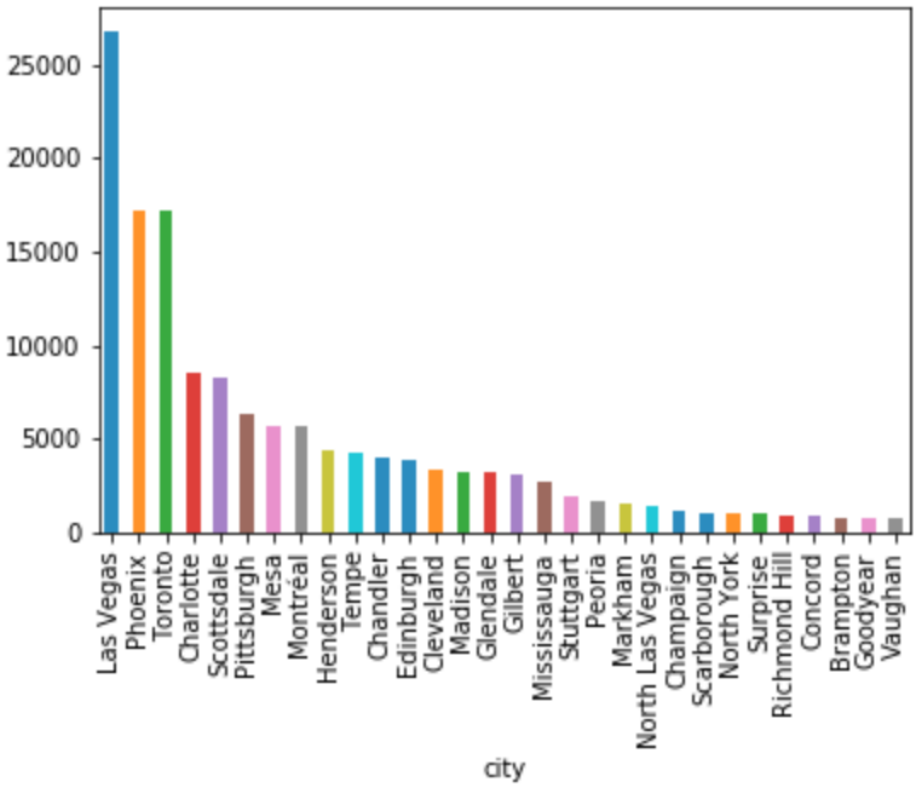

Figure 11: Number of Registered Businesses Per City.

Natural Language Processing(NLP) is a branch of Artificial Intelligence that deals with understanding and manipulating human language. To determine classify text according to context in the reviews, it is important for the machine first understand and grasp the language. In order to get useful data from the bulk of unstructured data, we apply various NLP techniques. Human language is difficult to interpret because of a variety of reasons, some of these are listed below in the subsections. After data cleaning steps, the cleaned data is then converted to a Bag of Words which is described in the next section.

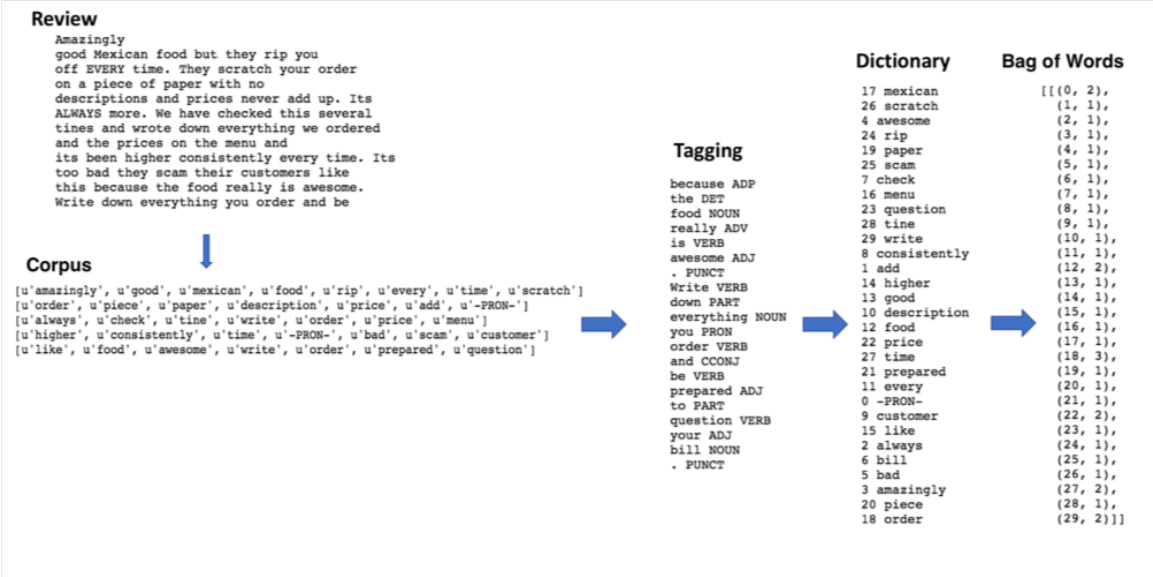

Figure 12: Workflow

5.2.1 Misspelling or alternately spelled words

Humans are prone to misspelling and alternately spelled words. It is common to write "you" as "u" or misspell homophonic words like "dessert" and "desert". Simple misspelling of homophonic words change the context of the word as well as sentences while alternately spelled words change .

5.2.2 Punctuation and Numerical Removal

With the help of Python library called SpaCy, all punctuations and numerical values are removed.

5.2.3 Language variability

Use of language changes with regions and so does sentence formation. Using Python library called pyEnchant, reviews only in English are filtered to be used for Topic Modeling.

5.2.4 POS Tags

Using Python library called SpaCy, POS like adverbs, pronouns are removed as such words do not give any useful insight.

5.3 Word to Corpus

The words from reviews are converted to numbers so that they can be interpreted by the ML models. All words are put into a dictionary data structure where key is the id of the word and value is the word itself. This helps keep track of words at a single place. The corpus is generated using the following methods listed in the subsections. The next steps are to apply these to machine learning models, covered in the next section.

5.3.1 Doc2Bow

Doc2bow is a simple method in Gensim [13] that converts documents to Bag of Words representation as shown in Figure 13.

5.3.2 Word Embedding using Word2vec

Word Embeddings [14] can be defined as a set of techniques in NLP for language modeling and feature learning. Words are embedded into multidimensional vector space in the form of numbers. These vectors aim to quantify similarities between linguistic items based on their distribution in the text. I have used Word2Vec [15] model created in Gensim [13] based on a paper by Google(change to names). It is a shallow 2-layer neural network. It is trained to construct similarities in word distribution. Words from each sentence are converted to tokens ans read sequentially. Words occurring in same or similar sentence structures over and over again are found

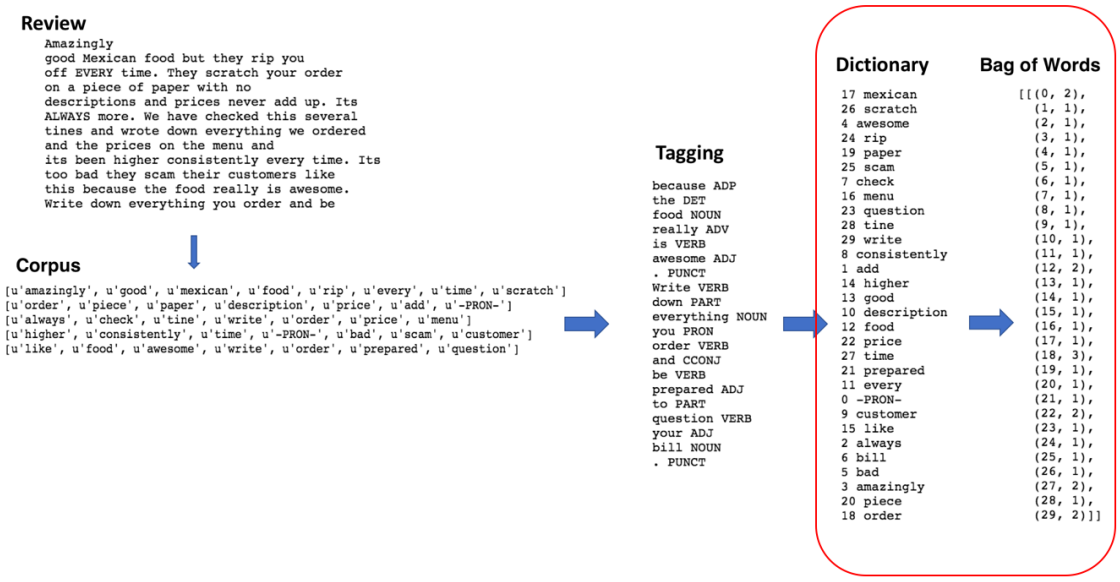

Figure 13: Bag of Words

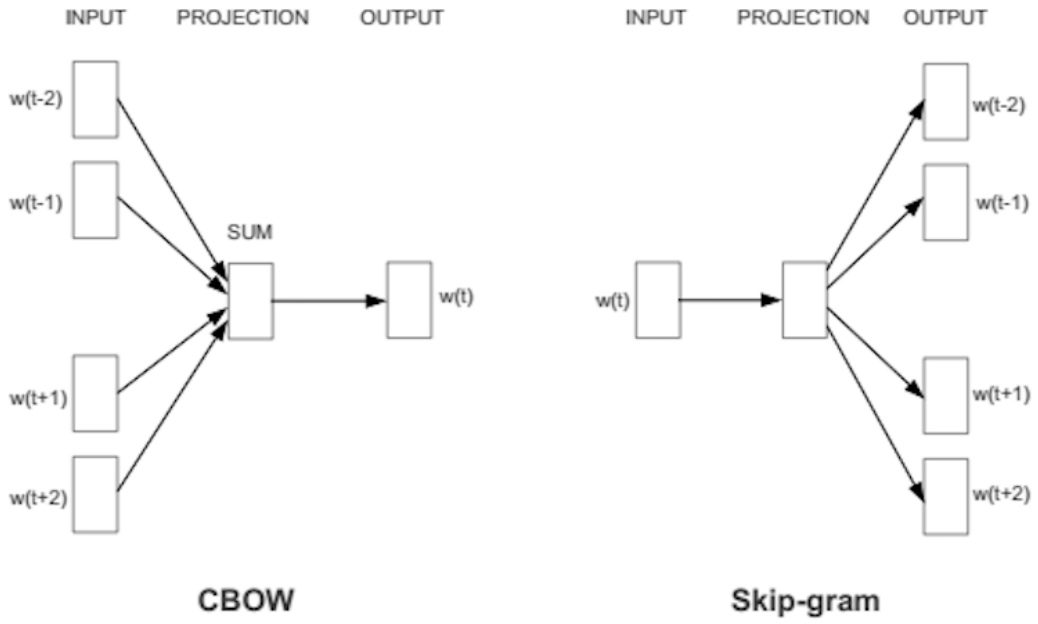

Figure 14: CBOW and Skip-Gram Methods

to have similar vectors. This means words appearing in similar context are close to each other in the vector space.

Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram [16]. In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words. According to the authors' note,CBOW is faster while skip-gram is slower but does a better job for infrequent words.

Gensim [13] uses skip-gram method to preserve the context of the word as well. The size of the context window determines how many words before and after a given word would be included as context words of the given word. According to the authors' note, the recommended value is 10 for skip-gram and 5 for CBOW

5.4 Machine Learning Models

Reviews do not have limit to the number of characters. This overwhelming number of reviews makes it difficult to analyze relevant reviews and their importance. Hence from the paper by Y. He, et al., the process of Topic Modeling and using Statistical Topic models PLSA and LDA was used for doing the same.

In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body. Intuitively, given that a document is about

a topic, one would expect words to appear in the document frequently. Unsupervised Learning techniques are used for Topic Modeling in this project.

5.4.1 Latent semantic analysis

Latent semantic analysis [17] is a technique to analyze relationship between documents based on a given set of concepts, it is similar to the other techniques presented where the assumption is that words having similar meaning of sorts would appear in similar texts thereby making documents similar. It approaches the problem by forming a tf-idf matrix of sorts, where we have rows representing words in the corpus and the columns representing the number of documents. It them beings by performing single value decomposition to produce a matrix that is similar to the original one, but has more sparse rows. Word vectors are then compared to each other using some kind of distance formula.

If we use cosine distance formula, a value of 1 would indicate that they are similar or otherwise different. LSA is similar to other techniques discussed in this section that it uses tf-idf matrix and some sort of distance formula to compare word vectors.

Some of the applications of LSA are: Finding terms similar to each other Given a search term, convert it to a lower dimensional space and find matching documents. Analyzing word associations in a word corpus.

5.4.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [5] is a Bayesian generative model for text. It is used as a topic model to discover the underlying topics that are covered by a text document. LDA assumes that a corpus of text documents cover a collection of K topics. Each topic is defined as a multinomial distribution over a word dictionary

with |V| words drawn from a Dirichlet: βk *in Dirichlet*(η)

Each document from this corpus is treated as a bag of words of a certain size, and is assumed to be generated by first picking a topic multinomial distribution for the document *Dirichlet*(α). Then each word is assigned a topic via the distribution βd, and then from that topic k, a word is sampled from the distribution βk. θd for each document can be thought of as a percentage breakdown of the topics covered by the document.

LDA is an unsupervised model which generates summaries of topics in terms of the discrete probability distributions over words, and it further infers per-document distribution over topics. Our model adds another latent factor, the interest Observing the word co-occurrences, the model will enable the word to choose the rights, and thereby the corresponding topics. LDA then using a dictionary of words provided by generates by SLA and clusters similar tweets together. We can then find the common topic from among the tweets that are clustered together and return the same as a result to the user.

5.4.3 OnlineLDA

In the papers [18] by M. Hoffman, et. al. have applied OnlineLDA is the combination of online variational Bayes and LDA applied over it. This model can also work on streaming data. The model has online stochastic optimization with a natural gradient step, which we show converges to a local optimum of the VB objective function.

5.4.4 Hierarchical Dirichlet Processing

Hierarchical Dirichlet Processing (HDP) [19] is used for grouped data. It is a nonparametric approach for clustering data that can be seen in the forms of groups. For each group of these data, it uses a Dirchlet process sharing a base distribution which is also a Dirchlet process.

Since HDP is a non parametric generalization of LDA, it is a core component of the infinite Hidden Markov model.

5.4.5 K-Means

K-Means is a vector quantization method that is used for document clustering and text mining. The principle of K-Means is that points in a plane can be grouped together based on their distance from k given points. The algorithm follows as follow, initialize k given points in place. Now, use the input set of points and measure the distance from each of the given K points. Points closer to one of the centroid are assigned that and at the end of an iteration, we have points belonging to one of the centroids. This process goes on and on until the centroid don't budge. In this case, we say that we have found one of the optimum solutions and figure out what points belong to which cluster.

This technique can be used along side TF-IDF to perform document clustering. One of the primary ways of encoding documents is that they can be thought of TF-IDF vectors. If every document can be expressed as a TF-IDF vector, then they can be expressed in a plane. It is now possible to initialize k centroid in a plane and find the distance of each document from a centroid. At the end of the iteration, we now have a clustering of documents given.

K-Means is a NP-hard problem and it is not possible to obtain the global maximum or minima clearly. Hence, a lot of variations of this algorithm exists, and one of the more popular variation is algorithm by Lloyd. K-Means algorithm has been used often and extensively in the field of astronomy, computer vision and agriculture.

5.5 Libraries used for Data cleaning

5.5.1 PyEnchant

PyEnchant library is a spell check library built by Dom Lachowicz. The library works has numerous dictionaries for different languages and hence can be used to check if words belong to a particular language, which in our case is English.

5.5.2 SpaCy

SpaCy [20] is a carefully memory-managed CPython library that excels at large-scale information extraction tasks. Confirmed to be the fastest by various Independent research, it is used for Parts of Speech tagging, number and punctuation removal, lemmetization and stemming words from the corpus.

SpaCy has numerous linguistic features that can be used to extract features from text. Linguistic features like part-of-speech tags, dependency labels and named entities can be extracted using SpaCy. The POS tagger is a trained neural network based on linguistic knowledge to add useful information and suggests the possible tagging of POS of words. Words are tagged to nouns, pronouns, verbs, adverbs, adjectives, numeric, punctuation. It also tags words as True or False to STOP. Parsing through the corpus, we can simply remove words that are tagged for STOP, go further by adding tags of words we do not wish to have in the corpus. Noun objects are further broken down to subject noun, object noun, proper noun, etc. SpaCy

tags Named entities as well. It has been trained to recognize Person, Organization, Location, Language Date, Product, Event etc. This Named Entity Recognition can also be used to gain topics related to Locations, food items, etc for Yelp reviews.

5.5.3 Gensim

Gensim library is a CPython library that implements various topic modeling models like LDA, LSA, HDP. It also has various packages that help in data cleaning and pre processing.

5.6 Visualization

LDA visualization is done using pyLDAvis library. pyLDAvis was used to plot the visualization presented in this documentation. It is based on a research paper presented at 2014 ACL Workshop on Interactive Language Learning, Visualization, and Interfaces. pyLDAvis offers two components for web interface that makes it possible for users to slice and dice LDA models. The first feature that it enables, is for users to select a topic and see what terms belongs to it. It also provides a slider, by which users can slide to increase or decrease the lambda value, which toggles the relevance factor for terms.

The second core feature of pyLDAvis is the ability to select a term (by hovering over it) to reveal its conditional distribution over topics. This distribution is visualized by altering the areas of the topic circles such that they are proportional to the term-specific frequencies across the corpus.

When performing, LDA most of the common questions tend to be:

Question 1 What is the meaning of each topic?

Question 2 How prevalent is each topic?

Question 3 How do the topics relate to each other?

pyLDAvis aims to answer Question 2 and 3 by presenting a global view of the topic model. Centers are determined by computing the distance between topics while the areas of the circle represents the prevalence score of topics. In order to answer question 1, pyLDAvis represents horizontal bar charts as the second component which shows what word comprises that topic. A pair of overlaid bars represent both the corpus-wide frequency of a given term as well as the topic-specific frequency of the term.

CHAPTER 6

Experiments and Results

Initial loading and set up of the dataset is a time consuming task. The review.json file is 3.64 GB. It is not possible to run analysis of the whole document at once. It took 3 days to initially load the JSON contents to a variable in Jupyter Notebook after which the notebook crashed due to intensive memory usage. Hence, the document was converted to a Comma Separated Value (CSV). The csv also turned out to be huge again but this step made file handling more manageable.

6.1 Data Cleaning

The next step, involved Data Cleaning, and due to large volume of data, applying non-English word removal to lemmatization of each review and storing it to a corpus variable, became challenging and was overloading the notebook causing it to crash often. Hence, as the solution to this, the process of corpus text generation is applied to one document at a time and saving it to the corpus text. At the end of this step, we get a list of documents which is in turn made up of list of words. The bigram model is applied on top of it. There are many words like "chicken tikka", "large groups", etc. that generally occur in pairs and hence it made sense to calculate for these words in pairs. This also improved the model clustering. OnlineLDA is one of the models where we can send one document at a time and create the model for one document at a time, adding and adjusting LDA topics and probabilities one document at a time. But we need the above preporcessing for all the other models.

6.2 Methods in Experiments

Experiments are conducted using various combinations of methods to see which methods work the best. The methods included:

- LDA
  - Normal Data Cleaning using Doc2Bow
  - Normal Data Cleaning using Doc2Bow + TF-IDF
  - Data Cleaning based on POS using Doc2Bow
  - Data Cleaning based on POS using Doc2Bow + TF-IDF
- LSI
  - Normal Data Cleaning using Doc2Bow
  - Normal Data Cleaning using Doc2Bow + TF-IDF
  - Data Cleaning based on POS using Doc2Bow
  - Data Cleaning based on POS using Doc2Bow + TF-IDF
- HDP
  - Normal Data Cleaning using Doc2Bow
  - Normal Data Cleaning using Doc2Bow + TF-IDF
  - Data Cleaning based on POS using Doc2Bow
  - Data Cleaning based on POS using Doc2Bow + TF-IDF
- OnlineLDA
  - Normal Data Cleaning using Doc2Bow

  - Normal Data Cleaning using Doc2Bow + TF-IDF
  - Data Cleaning based on POS using Doc2Bow
  - Data Cleaning based on POS using Doc2Bow + TF-IDF
- K-Means
  - Normal Data Cleaning using Word2Vec
  - Data Cleaning based on POS using Word2Vec

The reviews data was extracted based on the following criteria:

  - Yelp Dataset
  - Reviews for various categories
    * Restaurants in Yelp Dataset
    * Hotels in Yelp Dataset
  - "Panda Express" reviews on Yelp
  - Reviews for restaurants in Las Vegas on Yelp

6.3 Whole Yelp

The experiments were conducted for four sets of reviews. For the first part, 20000 reviews from across the whole Yelp dataset were taken at random. The review text was preprocessed using stop word removal, stemming-lemmatization and clustered into K=50 topics. Table 1, 2 show the top 5 words occurring in topics created using LDA, LSI and K-Means. I have further clustered them into set of 5 topics that seem to appear in the models. We observe that topics like Service, Hotels, Food items and Payment, Crowd are greatly discussed.

Table 1: Topics using K-Means

Service	Food items	Hotel	Food Texture
staff_friendly	ice_cream	water_heater	crispy_outside
friendly_attentive	yogurt	flooring	undercooked
fast_friendly	cheesecake	night	smooth
receptionist	burger	place	crispy
thank	sushi	stay	organic
knowledgeable	pizza	bathroom	spicy

Table 2: Topics using K-Means

Payment	Drinks	Crowd
credit	wine	busy
purchase	tea	rush
buy	beer	reservation
insurance	coffee	line_door
receipt	bottle	friday
discount	cocktail	large_group

Table 3 shows the Topics and words appearing in each of these words. In LDA, HDP and OnlineLDA, these topics are represented as a sum of all probabilities of words occurring in the topic. Hence saying 0.044*"room means Topic 1, probability of the word "room" occurring is 4.4%. For LSI, a positive probability means that the word will definitely appear in the topic and its probability will be the number given and negative probability indicates that the probability of the word not occurring in the Topic is the number given.

Table 4 shows the Topics and words appearing in each of these words.

Table 5 shows the Topics and words appearing in each of these words.

Table 6 shows the OnlineLDA Topics and words appearing in each of these words.

Table 3: 5 largest Topics in LDA and top 10 words in each

Topic #	Description
Topic 1	u'0.044*"room" + 0.027*"hotel" + 0.017*"stay" + 0.013*"good" + 0.011*"place" + 0.008*"nice" + 0.008*"bathroom" + 0.007*"night" + 0.007*"look" + 0.007*"bed"'
Topic 2	u'0.018*"place" + 0.014*"love" + 0.012*"great" + 0.012*"casino" + 0.010*"feel" + 0.010*"room" + 0.009*"bar" + 0.008*"beautiful" + 0.008*"fun" + 0.008*"vegas"'
Topic 3	u'0.018*"flight" + 0.014*"time" + 0.013*"fly" + 0.012*"get" + 0.011*"airline" + 0.009*"plane" + 0.009*"go" + 0.008*"seat" + 0.008*"phoenix" + 0.006*"way"'
Topic 4	u'0.023*"pie" + 0.020*"tooth" + 0.018*"thai" + 0.015*"furniture" + 0.012*"staff_attentive" + 0.012*"pizza" + 0.011*"gold" + 0.011*"garlic" + 0.009*"superb" + 0.009*"post_office"'
Topic 5	u'0.026*"food" + 0.025*"good" + 0.024*"place" + 0.021*"drink" + 0.016*"come" + 0.016*"order" + 0.013*"table" + 0.012*"time" + 0.012*"service" + 0.011*"bar"'

6.3.1 Visualization

The visualization for LDA using pyLDAvis is as follows.

We start off by sampling all Yelp reviews. We find that for K = 50, we have the following findings: Figure 15 shows the top 30 words in all.

Figure 16 shows the top 30 words for Topic 11. Figure 17 shows the top 30 words for Topic 8. Figure 18 shows the top 30 words for Topic 13. Figure 19 highlights clusters that contain the word "sushi".

Figure 20 highlights clusters that contain the word "service".

6.4 Restaurants

Table 7 shows LDA topics for Restaurant. Even though words in LDA topics are similar to that using K-Means, K-Means gives the best result in comparison. Table 8

Table 4: 5 largest Topics in HDP and top 10 words in each

Topic #	Description
Topic 1	u'0.001*sea_bass + 0.001*oyster + 0.000*ph + 0.000*jealous + 0.000*spontaneous + 0.000*after_read + 0.000*this + 0.000*corner + 0.000*not + 0.000*blackhead + 0.000*tendered + 0.000*so + 0.000*sickly + 0.000*unforgivably + 0.000*come + 0.000*food + 0.000*crinkle_cut + 0.000*consistent + 0.000*partridge + 0.000*answer'
Topic 2	u'0.000*unadventurous + 0.000*star + 0.000*cilantro_salsa + 0.000*piece + 0.000*melissa + 0.000*ship + 0.000*kiwi + 0.000*climbing + 0.000*maxwell + 0.000*crinkle + 0.000*wizard + 0.000*jewelry + 0.000*shortlist + 0.000*give + 0.000*good + 0.000*stet + 0.000*share + 0.000*mimosas + 0.000*sat')
Topic 3	u'0.001*markham_station + 0.000*wonk + 0.000*kart + 0.000*doc + 0.000*residence + 0.000*tossed + 0.000*personal + 0.000*hungover + 0.000*pier + 0.000*airport_hotel + 0.000*roasted_red + 0.000*eggs_benedict + 0.000*takeaway + 0.000*will + 0.000*experienced + 0.000*tate + 0.000*safe_bet + 0.000*great + 0.000*normal'
Topic 4	u'0.001*food + 0.001*good + 0.001*disheartened + 0.001*order + 0.001*place + 0.001*come + 0.001*roll + 0.000*tracie + 0.000*great + 0.000*time + 0.000*love + 0.000*check + 0.000*sushi + 0.000*delicious + 0.000*softball + 0.000*try + 0.000*need + 0.000*fresh + 0.000*go + 0.000*have'
Topic 5	(2, u'0.014*food + 0.011*not + 0.008*order + 0.008*good + 0.007*time + 0.005*panda + 0.005*go + 0.005*get + 0.005*panda_express + 0.005*place + 0.005*come + 0.005*location + 0.004*chicken + 0.004*ask + 0.004*employee + 0.004*want + 0.004*like + 0.004*orange_chicken + 0.004*service + 0.003*bad')

contains topic clusters for reviews about Restaurants in general. Restaurant being the most reviewed category in the dataset has the biggest chunk of reviews. Topic Models were run on 20000 reviews sampled at random for K = 50 clusters. Topics generally occurring in these review texts included "food", " dessert", "service", "time". Similar to results for overall Yelp reviews, these clusters have positive, negative and neutral experiences for each topic clustered together.

Table 5: 5 largest Topics in LSI and top 10 words in each

Topic #	Description
Topic 1	u'0.200*"time" + -0.197*"atmosphere" + -0.194*"have" + -0.189*"will" + 0.174*"-PRON-" + -0.163*"ice_cream" + 0.156*"die" + 0.156*"people" + -0.152*"super" + -0.150*"little"'
Topic 2	u'-0.654*"time" + 0.339*"great" + 0.224*"nice" + 0.201*"tell" + 0.190*"check" + 0.182*"go" + -0.180*"place" + -0.156*"room" + 0.153*"pool" + 0.140*"night"'
Topic 3	u'-0.559*"night" + 0.394*"pool" + -0.291*"want" + -0.210*"pay" + -0.184*"nice" + 0.162*"minute" + 0.155*"wait" + -0.129*"service" + 0.127*"walk" + -0.118*"tour"'
Topic 4	u'0.338*"love" + -0.316*"breakfast" + -0.258*"amazing" + -0.202*"white_castle" + 0.197*"service" + 0.195*"price" + -0.181*"delicious" + -0.181*"coffee" + 0.166*"food" + -0.156*"slider"'
Topic 5	u'0.389*"drink" + 0.300*"awesome" + 0.210*"amazing" + 0.195*"this_place" + 0.188*"coffee" + -0.184*"breakfast" + -0.165*"beer" + -0.158*"ride" + -0.154*"great" + 0.133*"place"'

Table 6: 5 largest Topics in OnlineLDA and top 10 words in each

Topic #	Description
Topic 1	u'0.045*"medium" + 0.032*"tender" + 0.032*"new_york" + 0.027*"refill" + 0.026*"spinach" + 0.022*"presentation" + 0.020*"second_time" + 0.019*"nicely" + 0.018*"crisp" + 0.017*"goat"'
Topic 2	u'0.060*"dentist" + 0.048*"house" + 0.035*"tooth" + 0.034*"cleaning" + 0.022*"sub" + 0.022*"split" + 0.019*"image_dental" + 0.017*"convenient" + 0.017*"roasted" + 0.015*"ounce"'
Topic 3	u'0.024*"interior" + 0.022*"unique" + 0.020*"crab_dip" + 0.015*"totally_worth" + 0.015*"craft_beer" + 0.015*"friendly_staff" + 0.014*"restroom" + 0.014*"ever" + 0.013*"shave" + 0.011*"wind"'
Topic 4	0.158*"good" + 0.150*"food" + 0.120*"great" + 0.086*"service" + 0.086*"place" + 0.048*"love" + 0.031*"try" + 0.029*"come" + 0.019*"awesome" + 0.019*"time"'
Topic 5	u'0.156*"hotel" + 0.032*"bathroom" + 0.024*"bread" + 0.023*"dollar" + 0.020*"mix" + 0.019*"do" + 0.018*"leg" + 0.015*"four_seasons" + 0.015*"section" + 0.014*"level"'

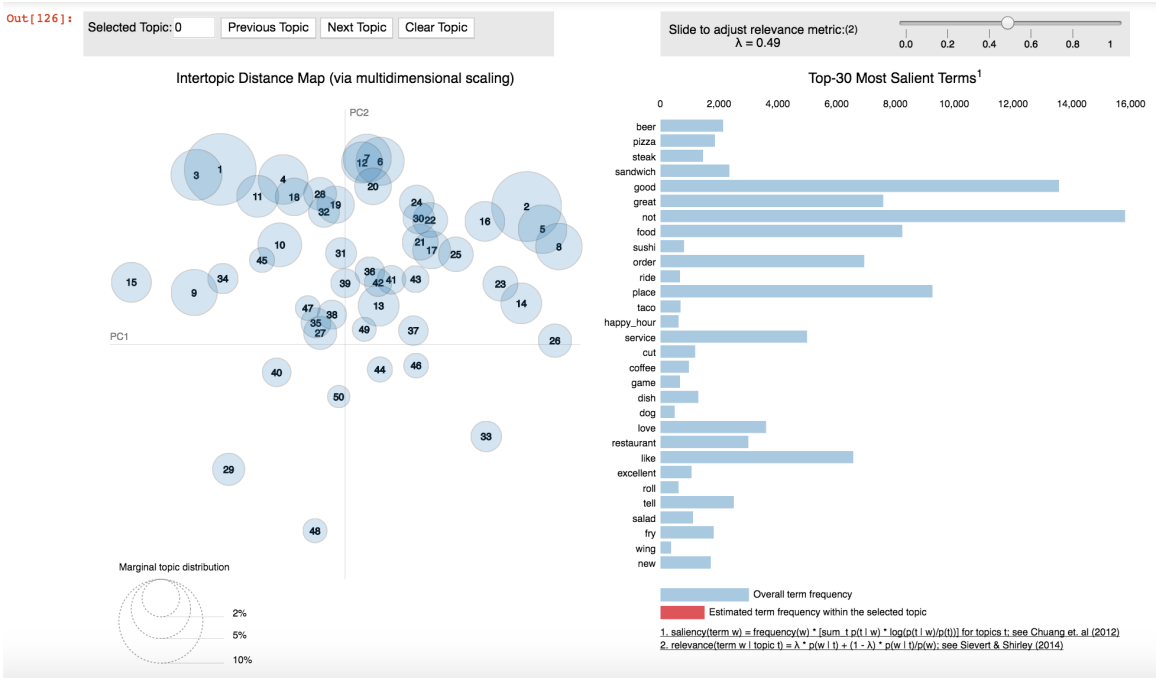

Figure 15: LDA Topic Model

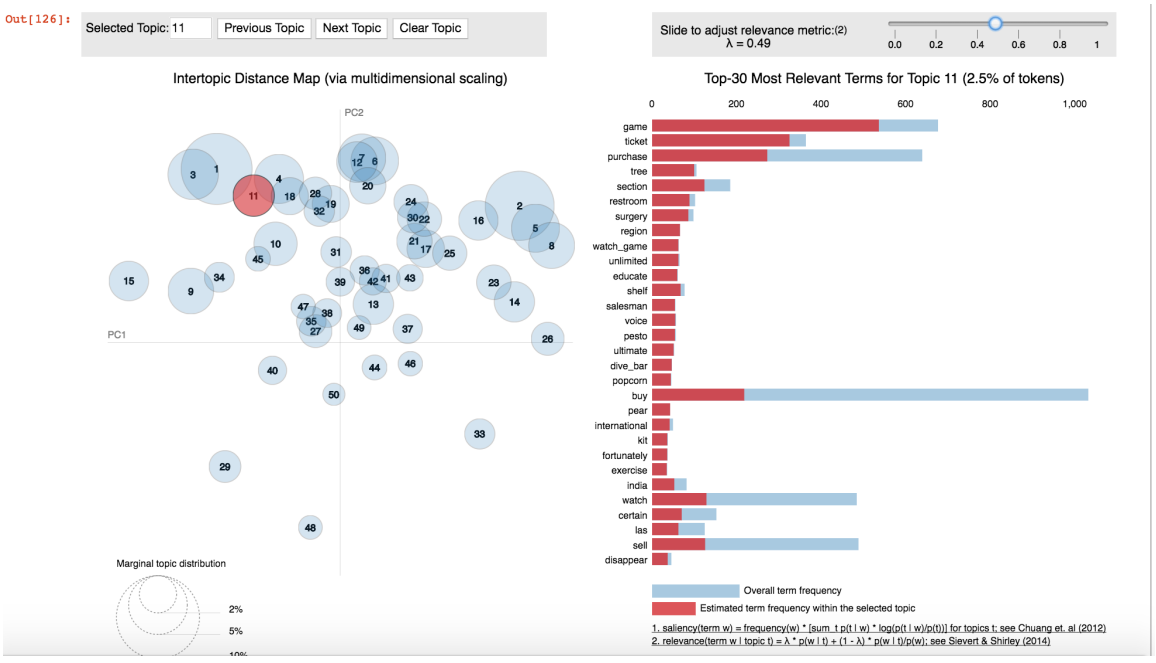

Figure 16: LDA Topic Model for Cluster 11

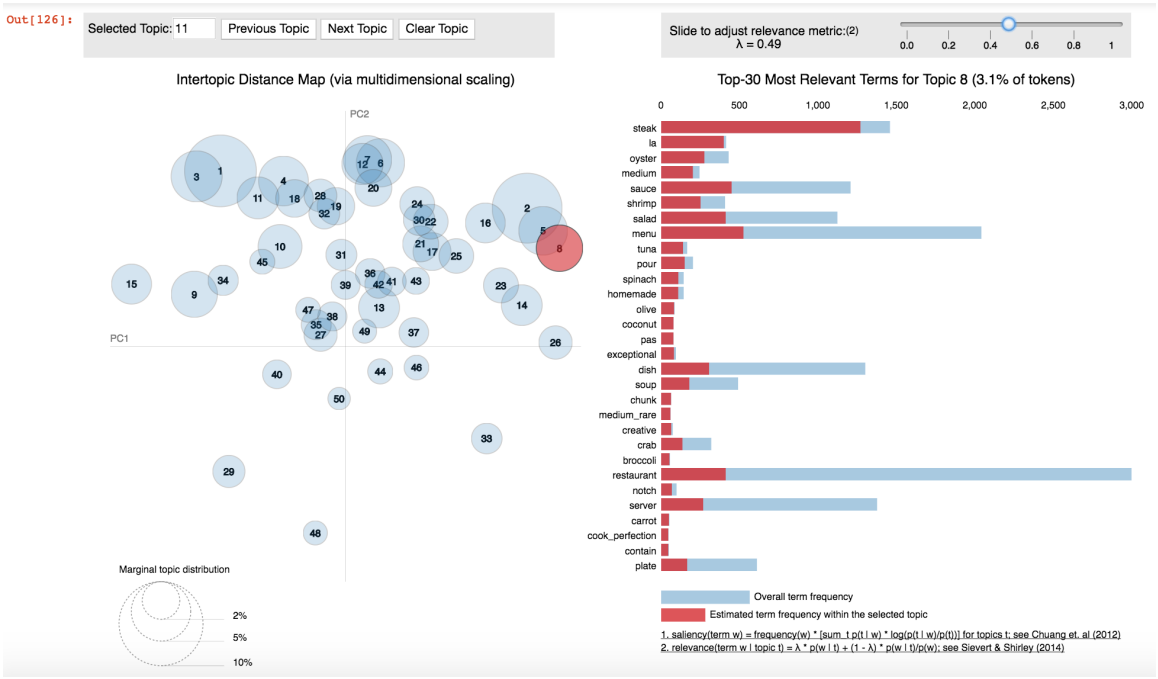

Figure 17: LDA Topic Model for Cluster 8

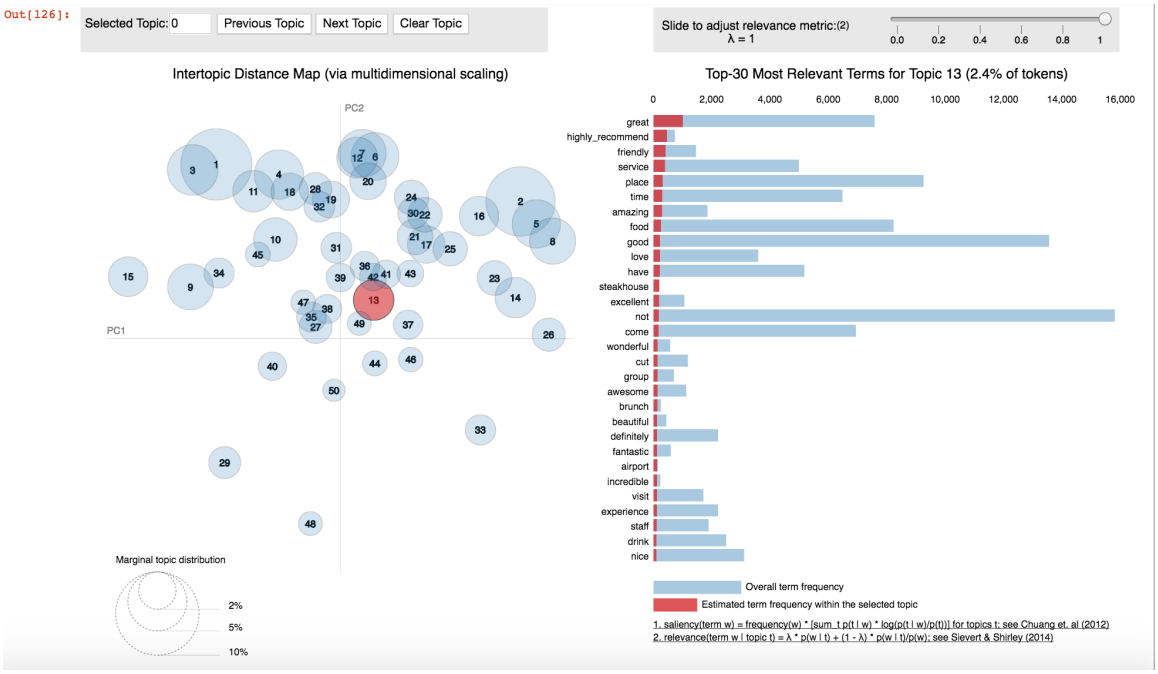

Figure 18: Topic Modeling

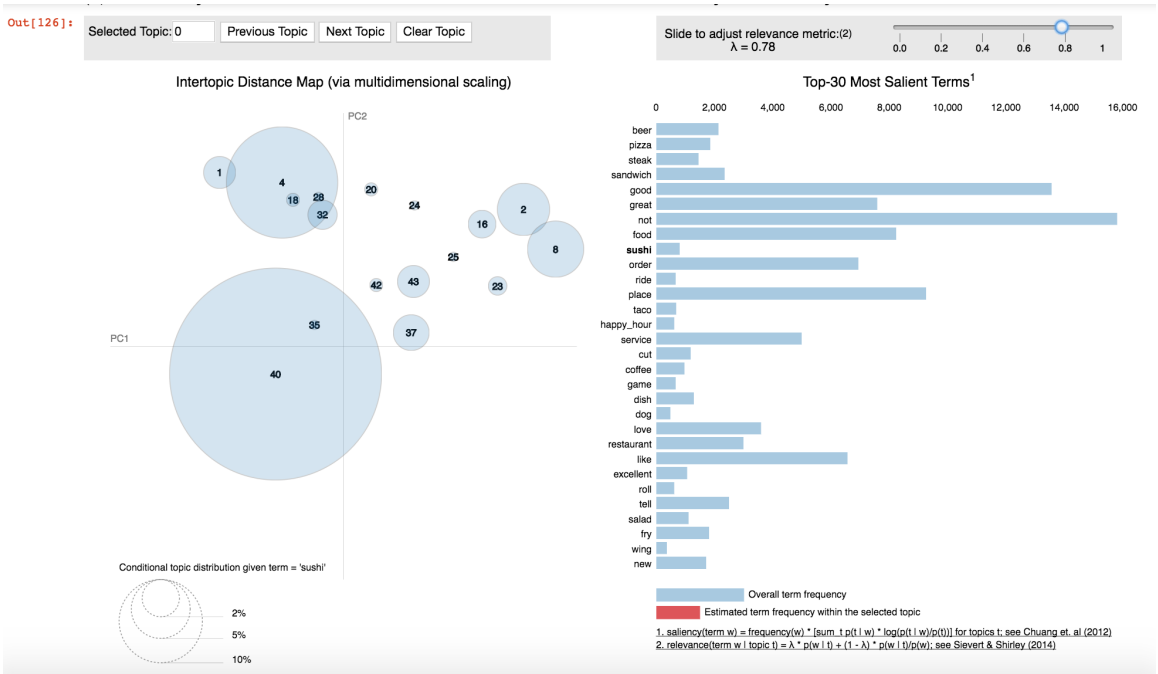

Figure 19: LDA topics containing word "sushi"

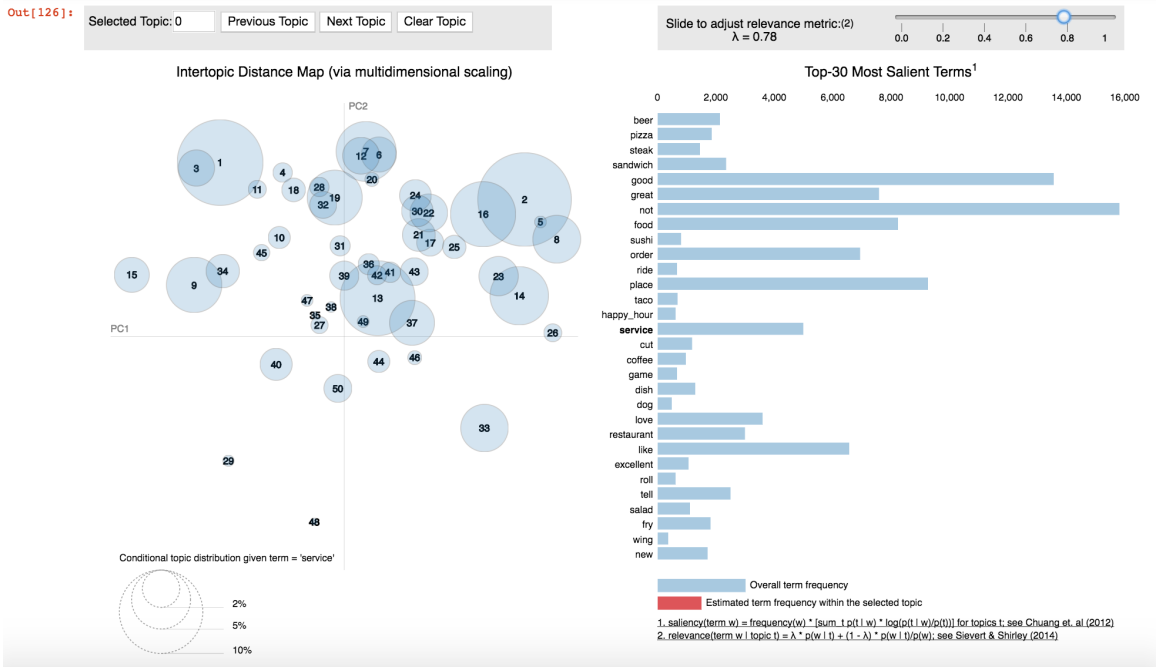

Figure 20: LDA topics containing word "service"

Table 7: 5 largest Topics in LDA and top 10 words in each for Restaurants

Topic #	Description
Topic 1	u'0.056*"fry" + 0.034*"vegan" + 0.031*"good" + 0.024*"cheese" + 0.021*"burger" + 0.018*"bbq" + 0.016*"not" + 0.015*"great" + 0.015*"sauce" + 0.014*"order"'
Topic 2	u'0.007*"duck" + 0.006*"steak" + 0.006*"bowl" + 0.005*"good" + 0.005*"order" + 0.005*"food" + 0.005*"high_end" + 0.005*"vegetable" + 0.005*"rib" + 0.005*"spicy"'
Topic 3	u'0.057*"sushi" + 0.032*"cut" + 0.028*"good" + 0.016*"chicken" + 0.016*"place" + 0.011*"spice" + 0.011*"love" + 0.010*"pork_belly" + 0.010*"go" + 0.009*"chunk"'
Topic 4	u'0.032*"time" + 0.019*"food" + 0.017*"service" + 0.016*"mac" + 0.013*"good" + 0.013*"present" + 0.013*"have" + 0.012*"know" + 0.012*"dining_experience" + 0.011*"attitude"'
Topic 5	u'0.143*"big" + 0.070*"business" + 0.048*"use" + 0.040*"vegetarian" + 0.039*"tea" + 0.030*"crab" + 0.029*"corner" + 0.026*"dine" + 0.023*"pie" + 0.023*"small"'

Table 8: Topics using K-Means on Restaurants

Service	Menu items	Food Texture	Order
staff_friendly	chicken_breast	spicy	order
absolutely_love	sushi	good	table
deliver	stake	like	come
attitude	cut	fry	wait_staff
table	beer	vegan	serve_size
knowledgeable	sandwich	plenty	wait

6.5 Hotels

Table 9 contains topic clusters for reviews about Hotels in general. Restaurant being the second most reviewed category in the dataset also has the biggest chunk of reviews(almost 90 thousand). Topic Models were run on 20000 reviews sampled at random for K = 50 clusters again. Topics generally occurring in these review texts included "rooms", "hotel", "food", "drinks". Similar to results for overall Yelp

reviews, these clusters have positive, negative and neutral experiences for each topic clustered together.

Table 9: Topics using K-Means on Hotels

Service	Rooms	Hotel	Food	Drinks
friendly	water_heater	interior	order	wine
friendly_attentive	very	flooring	purchase	rush
attentive	TV	lobby	restaurant	wine
receptionist	heater	lit	try	busy
thank	like	hotel	burger	friday

6.6 Panda Express

Table 10 contains topic clusters for reviews about "Panda Express". The restaurant has around 2000 reviews on Yelp and we use them all to create topic Models for K = 5 clusters again. Reviews talk about the "service", "location" and "food items" like orange_chicken and chicken the most. There are topics that have positive and negative words associated with various topics indicating the positive or negative reviews for the same respectively. The topic models of a few other restaurants have similar results, sometime including the "time to serve" as well.

Table 10: Topics using K-Means on Panda Express

Service	Location	Food Items
order	place	food
slow	come	order
write	fresh	like
not	panda	panda_express
good	pretty	chicken
time	drive	orange_chicken

6.7 Restaurants in Las Vegas

Finally the topic modeling on Restaurant reviews in Las Vegas(Figure 11) showed that the customers were the most interested in "drinks", "food", "service", "hotel" topics. Since Las Vegas has the most number of reviews and restaurants registered on Yelp as indicated in Figures 9 and 10, we took a subset of 20000 reviews and split the corpus into 50 reviews.

Table 11: Topics using K-Means on Las Vegas Restaurants

Service	Food items	Hotel	Drinks
outstanding	chicken_fry	room	wine
service	lobster	order	tea
time	drinks	music	beer
room	chicken	blackjack	coffee
nice	sauce	buffet	bottle
reservation	dessert	discount	cocktail

The K topics figure was taken from the paper [8]. The experiments conducted in the paper indicates that K=50 gives optimal number of topics to avoid under fitting or over fitting.

6.8 Other Observations

6.8.1 Reference based topic context

As we infer about the differences in the categories, we observe that the two topic model results for restaurants and hotels differ in the topics and contents of the topic. The words in "Service" topic of both the models differs with respect to the categories and the models pick those up well. Table 13 has a few examples of the categorical Service words.

6.8.2 Positive and Negative sentiments in Subtopic

When we dive down further in the analysis, we find that these clusters actually contain positive and negative sentiments attached to each cluster. For example, as seen in the Table 12 few clusters that talk about "Service" have negative or positive or neutral words like "good", "great", "bad", "late", "just_enough" etc. clustered together.

Table 12: Positive and Negative Service Topics

Sentiment	Topics
Positive	u'0.091*"great" + 0.049*"place" + 0.043*"food" + 0.042*"service" + 0.038*"amazing" + 0.032*"love" + 0.025*"awesome" + 0.024*"good" + 0.020*"recommend" + 0.016*"try"'
Negative	u'0.287*"have" + 0.266*"service" + 0.265*"price" + 0.215*"not_recommend" + 0.211*"staff" + 0.205*"eat" + 0.173*"order" + 0.142*"drink" + 0.135*"not"'

Table 13: Example of Service in Categories context

Category	Topics
Hotels	u'0.624*"place" + 0.370*"car" + 0.361*"time" + 0.279*"great" + 0.274*"good" + 0.193*"service" + 0.135*"look" + 0.083*"tour" + 0.078*"hotel" + 0.077*"walk"'
Restaurants	u'0.257*"friendly" + 0.256*"this_place" + 0.191*"food" + 0.173*"crunchy" + 0.172*"nice" + 0.150*"fun" + 0.145*"very" + 0.143*"store" + 0.142*"service" + 0.136*"this"'

6.9 Model Comparison

6.9.1 Coherence score

Models can also be compared with each other using Coherence score. Figure 21 compares Coherence scores for LSI, HDP, LDA and OnlineLDA for the first dataset.

The above method can not be used with K-Means as it only generates clusters and Coherence model also uses probabilities of word occurrence in topics for comparison,

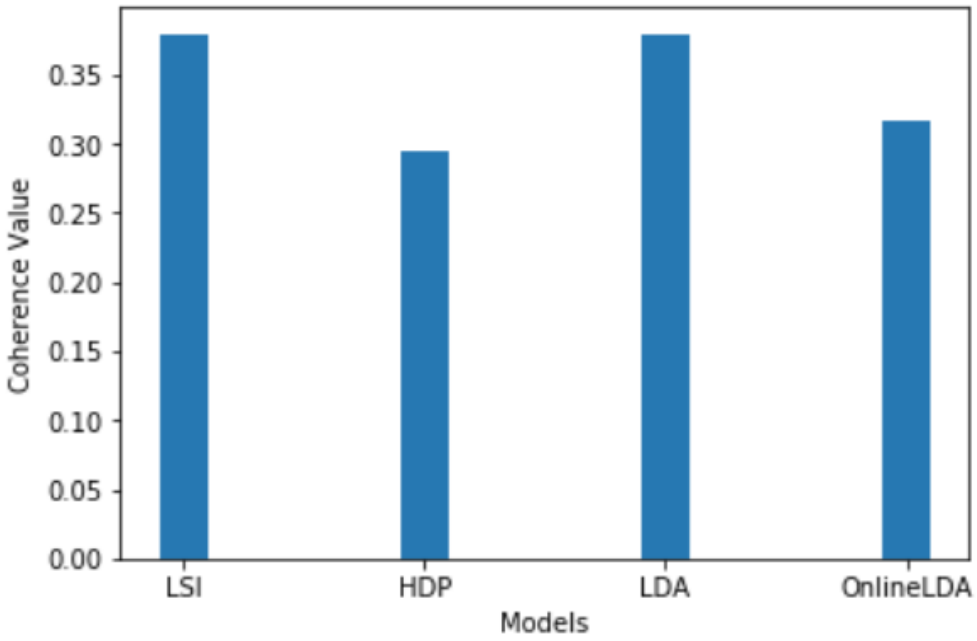

Figure 21: Coherence score for LSI, HDP, LDA and OnlineLDA

the models are judged based on human feedback. From the topic models generated above, we find that POS-tagged corpus based K-Means has the best topic distribution among all the models. POS tagged TF-IDF LDA model has the second best topic distribution. Word Embeddings are best able to retain the context and semantic structure of words hence can cluster the words better into topics.

6.9.2 Time to implement

Although, K-Means has the best clustering, it takes up a lot of time to give back results. Word Embeddings used to cluster using K-Means do an excellent job of retaining the context of words but also considers their frequency of occurrence in the text. The vector size is set to be 32, with words occurring atleast 5 times in the corpus, this helped reduce the time taken to run the model. Initial run for the

Table 14: Time taken for 20000 reviews, K=50

Topic Model Technique	Time taken
LDA(with POS and TF-IDF)	153.6 s
LSA(with POS and TF-IDF)	20.4 s
HDP(with POS and TF-IDF)	219.5 s
OnlineLDA(with POS and TF-IDF)	146.5 s
K-means(with POS)	16227 s

Table 15: Time taken for 2000 reviews, K=5

Topic Model Technique	Time taken
LDA(with POS and TF-IDF)	8.6 s
LSA(with POS and TF-IDF)	0.4 s
HDP(with POS and TF-IDF)	16 s
OnlineLDA(with POS and TF-IDF)	5.9 s
K-means(with POS)	25 s

20000 reviews across the whole Yelp Dataset took almost 5 hours to return with the result. Reducing the vector dimensions reduced the time by 60% with some loss of vector information. While, LDA, HDP and OnlineLDA models tend to take moderate amount of time. LSI, though consumes the least amount of time in all, has clusters that are not very descriptive. Among all the types of cleaning and experimentation, we get best results using data that have specific POS tagged words and TF-IDF over various documents. Table 14 and Table 15 shows the time taken by each of the techniques. As we see, time taken for a smaller dataset is significantly less. K-Means works perfectly for smaller datasets and can be used on a machine with more power or run in parallel or distributed environment to get best possible results.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

Understanding context within text is hard, and it largely depends on the data being fed to the algorithm. The underlying assumption that review text within Yelp follow a certain theme is true for most cases when comparing with the results from this research. Thus, with certainty we can say that if a review continues in a positive sentiment, it will go on to identify the underlying features that make it so. Understanding this through a ML model is tricky because of computational constraints on training the model and hence cleaning strategies become a pivotal part of the process.

As observed, customers and businesses of Las Vegas are the most active amongst all other cities. Topics such as "service", "wait_time", "happy_hour" are discussed often over all. ML techniques can effectively extract these subtopics and present a structured view of what it means to look at a particular business. With these results, it becomes possible to identify new topics on the fly using OnlineLDA, hence detecting fake or solicited reviews.

Among the ML techniques employed, word embeddings prove to be more efficient in generating subtopics. Although effective, K-Means is a very computationally heavy process if there are too many reviews. Combining data cleaning methods like TF-IDF and POS tagging work as good as K-Means with word embeddings for larger datasets and are not as computationally expensive.

7.2 Future Work

There are various other methods to get word embeddings from the review Corpus like available in Gensim can also be tested with. Also there is a need for a good metric to judge NLP models described in this report. As seen, the current methods are black boxes and cannot be taken as it is.

LIST OF REFERENCES

[1] D. Dai, G. Z. Jin, J. Lee, and M. Luca, "Aggregation of consumer ratings: An application to yelp. com," 2017.

[2] "Topic model image," https://www.researchgate.net/figure/Informal-explanation-of-the-intuition-behind-topic-modelling-adapted-from-Blei-2012_fig3_264656298, accessed: 2018-04-30.

[3] "Yelp 15 things to know," https://www.searchenginejournal.com/15-things-may-not-know-yelp/129758/, accessed: 2018-01-30.

[4] "Topic modeling flow," http://chdoig.github.io/pytexas2015-topic-modeling/#/, accessed: 2018-04-30.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] 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*. Association for Computational Linguistics, 2010, pp. 100–108.

[7] "Coherence in topic models," http://qpleple.com/topic-coherence-to-evaluate-topic-models/, accessed: 2018-04-30.

[8] J. Huang, S. Rogers, and E. Joo, "Improving restaurants by extracting subtopics from yelp reviews," *iConference 2014 (Social Media Expo)*, 2014.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[10] J. Linshi, "Personalizing yelp star ratings: A semantic topic modeling approach," *Yale University*, 2014.

[11] D. Hazarika, S. Poria, P. Vij, G. Krishnamurthy, E. Cambria, and R. Zimmermann, "Modeling inter-aspect dependencies for aspect-based sentiment analysis," https://naacl2018.wordpress.com/2018/03/02/list-of-accepted-papers/, accessed: 2018-01-30.

[12] "Yelp dataset image," https://www.yelp.com/dataset, accessed: 2018-04-30.

[13] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[14] Word Embedding Wikipedia contributors, "Word embedding — Wikipedia, the free encyclopedia," 2018, [Online; accessed 3-May-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Word_embedding&oldid=836044700

[15] Word2Vec Wikipedia contributors, "Word2vec — Wikipedia, the free encyclopedia," 2018, [Online; accessed 3-May-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Word2vec&oldid=836426989

[16] "Word2vec image," https://deeplearning4j.org/word2vec.html, accessed: 2018-04-30.

[17] LSA Wikipedia contributors, "Latent semantic analysis — Wikipedia, the free encyclopedia," 2018, [Online; accessed 3-May-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Latent_semantic_analysis&oldid=835635717

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

[19] HDP Wikipedia contributors, "Hierarchical dirichlet process — Wikipedia, the free encyclopedia," 2018, [Online; accessed 3-May-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hierarchical_Dirichlet_process&oldid=829917828

[20] "Spacy," https://spacy.io/, accessed: 2018-04-30.