Nidhi Trivedi A20410681

Final Project Report

LDA & Word2Vec for semantic Analysis

Abstract:

Yelp is a decent wellspring of surveys for clients, which enable them to pick the best business in the area. The dataset is a subset of businesses, users, tip, photos, checkin and review data which provides a platform to learn about various techniques like Data Mining, NLP, ML etc. to discover patterns, analyse textual data and find meaningful information. This project will focus on providing an overview of how can LDA and Word2Vec can be used for Semantic Analysis. Topic modelling techniques were developed to discover different topic dimensions of a document set. This paper will document the evaluation of two different techniques LDA and Word2Vec on yelp dataset & Enron Email Dataset followed by analysing the results obtained to learn which technique is better for semantic analysis and which is better for sentiment analysis and why.

I) Introduction:

A semantic analysis is intended to discover the meaning of structured and relevant information from the data. Semantic search provides more meaningful search results by evaluating and understanding the search phrase and finding the most relevant results. The goal of the project is to implement Latent Dirichlet Allocation (LDA) and Word2vec techniques for Semantic search on Yelp dataset and Enron Email dataset. Based on results from these 2 techniques, analyse which of these techniques is suitable for Semantic analysis and which one is for Sentiment analysis.

II) Techniques/Approaches:

Latent Dirichlet Allocation (LDA): It is one of the most common topic modelling techniques. It can be understood via 2 principles. First, every document is a mixture of topics and second, every topic is a mixture of words. So basically, LDA is used to estimate both the principles at the same time. LDA will be useful in determining which words exactly belong to a specific topic and then it will determine the topic of document by then examining its probabilities. In order to work with LDA, document term matrix is required. So, first a document term matrix, which is a matrix describing the frequencies of terms that occurs in a corpus, will be calculated. In DTM, rows correspond to documents and columns corresponds to terms.

Word2Vec: The word2vec technique is used to produce word embeddings. It is a tool which provides an efficient implementation of the continues Bag-Of-Words and Skip-gram architecture for computing vector representation of words. Word2Vec tool takes a large text corpus as input & produce the word vector as output. It first constructs a vocabulary from the training text data and learns vector representation of words. Each unique word in corpus is assigned a vector in space. These word vectors are positioned in such a way that words that share common contexts in corpus are in the closer proximity to one another in the space. We are using it to find closet words for a user-specific words.

III) Packages and Dataset:

Programming Language/Packages: This project will be using R which provides a great environment for data mining, statistical computing and graphics. R can be easily extended by using available packages at CRAN, thus making implementation easy.

Packages - the packages which will be used are : Text Mining(tm), Natural Language Processing(NLP), Wordcloud, jsonlite, Latent Semantic Analysis(lsa), RTextTools, ggplot2, SnowballC, e1071(For latent class analysis, Naive-Bayes classifier, SVM etc.), Plotly, cluster, factoextra, topicmodels, word2vec, word_analogy.

Data Source:

Link : (https://www.yelp.com/dataset/challenge)

Yelp Dataset:

The dataset is obtained from 11th round of 'Yelp Dataset Challenge' and contains information about local businesses in 12 metropolitan areas across 4 countries, with around 156639 businesses and 5 million reviews from over 1.1 million users. The dataset downloaded is of 5.79 GB which contains 6 files in JSON format (user, business, review, checkin, tip, photo). This project will be focusing on 2 out of 6 files which are business.json and review.json. 100k reviews was taken from *Restaurant* category with *cuisines* as subcategory namely, Chinese, French, Indian, Italian and Mexican with 20k records each. 100k reviews was also used for the experiment.

Enron Email Dataset: This dataset is consist data from 150 users, mostly from the senior management of Enron Corporation. The dataset is basically took from the email communication of Enron Employees. Email were extracted and each mail were stored into different text file, total 137k emails were extracted and 100k records was used for the experiment.

IV) Approach:

Extraction: From both of the dataset total 100k records we took and store it into CSV and TXT format.

Pre-processing Steps:

Yelp dataset is originally in JSON format which will be first converted to .csv format before pre-processing it. In order to do that, library jsonlite was used. With the help of stream_in() I was able to read JSON data, which took hours to read it. Then the JSON data was flatten into regular 2 dimension tabular structure, which was then checked if it was a data frame object using as.data.frame(). Once I was sure data is ready, business_Id and categories was extracted from business.json and business_Id and text was extracted from review.json. I perform the following task for data pre-processing:

-Stopwords,stopwords_en, whitespace, numbers and punctuations were removed using the tm package which cleans the raw data and prepare data for further analysis.

-I created Document term matrix by using the inbuilt DocumentTermMatrix(). This matrix describes the frequency of terms which occurs in a document.

- dtm<-DocumentTermMatrix(corp, control = list(wordLengths=c(4, Inf)))

-Stemming was performed on the cleaned data because the classifier doesn't understand verbs and treats them as different words.

-Pruning words by frequency: words that occurs in very few documents were removed, which is good because the entire focus is on the words which have high frequency and it also reduces computation time.

-I also remove the sparse terms so that the whole pre-processing will be done.

V) Experiments:

Implementing LDA:

After pre-processing data and getting document term matrix in previous steps, we are now ready to perform LDA. In order to do so, we used Gibbs sampling for estimation. The main parameters for LDA are as follows:

- burnin - starting period, steps which does not reflect distribution property are removed.

- iter - number of iterations

- thin - number of iteration at which correlation between samples is avoided

- seed - an integer for each starting point

- nstart - number of runs at different start points

- best - return result of best run

LDA(dtm, 4, method="Gibbs", control=list(nstart= 5, seed = list(2003,5,63,100001,765), best= TRUE, burnin = 4000, iter = 1000, thin= 500))

Implementing Word2Vec:

We used inbuilt functions of Word2Vec. For work with Word2Vec we have to give text corpus as a input file and we will get binary file as output. As rword2vec package requires the single text file as an input for training a model, I had to combine all 100k text files into a single text file. This resulting file consists of all 20,000 reviews which were extracted.

- word2vec(train_file =" " , output_file =" " , binary=1)
- train_file: path of the text file which will be used as a training file to create the vocabulary
- output_file: path for the word2vec output file in a binary format

Parameter Tuning:

As should be obvious from above area, LDA capacities have a few parameters. I have utilized default parameters to get the outcome, yet couple of parameters can be tuned to get enhanced outcomes. Parameters, for example, iter, k, nstart can be tuned to get enhanced outcomes. Be that as it may, tuning them to the extraordinary can be awful as far as calculation.

Iter : in the event that we set it too high then it might bring about the great model yet it will cost more on calculation particularly in situations where corpus is immense.

nstart: like iter, if the esteem set too high then it will have more computational cost.

k: if the esteem is set too low then we won't not get every single dormant theme. On another hand, if the esteem is set as well high then we may make them astound subjects which may not bode well. The site without moving quality will be the no of classes or distinctive kinds of information in the corpus. We can simply perform examination on points made with various qualities and figure out which ought to be the site esteem.

VI) Results & Analysis:

Results for Yelp Dataset:

LDA executed for a different number of values for parameter k. k= 3,4,5,6,10,50,100 and 200 was used to get different results. I used 100k records for this.

Following table indicates the top 10 words for that topic when k=5 ,100 and 200.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	pizza	delici	chicken	salad	sauc
2	famili	fresh	dish	chees	tast
3	pasta	taco	indian	dessert	pretti
4	water	mexican	rice	bread	soup
5	person	awesom	spici	wine	portion
6	manag	super	buffet	vega	chines
7	italian	salsa	tast	plate	beef
8	waitress	burrito	flavor	appet	sweet
9	return	authent	naan	french	noodl
10	arriv	fish	curri	steak	flavor
label of class	Italian	Mexican	Indian	French	Chinese

For k=5

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	pretti	taco	chicken	tast	dish	wine	pizza	water	delici	dish
2	qualiti	mexican	indian	sauc	dessert	vega	salad	manag	fresh	soup
3	option	salsa	buffet	flavor	plate	french	chees	waitress	portion	rice
4	probabl	burrito	naan	bread	appet	steak	famili	arriv	tasti	chines
5	close	fish	curri	pasta	cours	wonder	hous	left	fill	beef
6	decent	authent	butter	meat	entre	reserv	free	return	huge	noodl
7	overal	bean	garlic	italian	husband	absolut	pick	bring	super	shrimp
8	chang	chip	lamb	sweet	main	fantast	week	brought	size	pork
9	mayb	fast	spice	bland	cream	select	stop	final	onion	roll
10	impress	awesom	spici	piec	chef	outsid	live	cold	decor	spici
Comments	Chinese	Maxican	indian	italian	Indian	french	italian	Maxican	French	chinese

For k=10

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20	Topic 21	Topic 22	Topic 23	Topic 24	Topic 25	Topic 26	Topic 27
1	insid	pizza	fast	cream	chang	hous	taco	green	tast	salsa	delici	qualiti	decor	portion	super	delici	choic	person	chicken	parti	leav	bread	couldnt	burrito	entre	dessert	steak
2	outsid	qualiti	fresh	dessert	regular	salad	salsa	bean	bland	chip	absolut	impress	ambianc	size	tasti	share	includ	hand	butter	care	bring	butter	finish	bean	appet	appet	vega
3	crispi	mayb	delici	chees	make	bread	bite	rice	flavor	mexican	high	overal	overal	huge	awesom	high	surpris	make	flavor	person	left	warm	stop	mexican	includ	main	strip
4	town	pick	super	light	hope	delici	wish	onion	hard	taco	like	fresh	especi	share	delici	huge	especi	probabl	amount	thank	return	garlic	bite	taco	warm	entre	ambianc
5	salad	hand	tasti	warm	name	fresh	fresh	amount	return	bean	vega	surpris	impress	pretti	huge	person	found	instead	strip	help	hope	cours	leav	salsa	yelp	share	french
6	pick	onion	huge	sweet	total	live	hous	like	town	warm	glass	hope	warm	week	high	portion	person	name	potato	left	coupl	piec	wish	chees	like	bread	outsid
7	includ	bite	amount	includ	instead	warm	green	regular	probabl	mayb	yelp	fantast	make	dessert	wish	coupl	tasti	cours	found	return	hand	fill	found	fill	water	extrem	onion
8	return	return	qualiti	green	bring	chees	butter	make	extra	decent	week	fill	help	chees	fast	return	regular	total	person	total	thank	vega	coupl	chip	delici	leav	fantast
9	light	thank	thank	surpris	portion	surpris	fast	sweet	includ	live	chees	high	fresh	fill	vega	bread	delici	overal	cream	surpris	final	dish	probabl	overal	amount	portion	main
10	garlic	chines	brought	hand	overal	share	chines	chees	surpris	fresh	piec	arriv	hard	chef	hand	fantast	select	finish	fresh	hard	wonder	mayb	total	make	chines	live	return

For k=100

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20	Topic 21	Topic 22	Topic 23	Topic 24	Topic 25
1	chef	sauc	wine	pizza	live	steak	chang	chicken	entre	noodl	meat	wrong	sauc	bean	curri	left	authent	sauc	sweet	total	fresh	delici	taco	delici	qualiti
2	prepar	chees	glass	italian	close	outsid	hope	hand	appet	soup	flavor	instead	green	rice	naan	arriv	mexican	hard	tast	especi	tasti	absolut	salsa	absolut	hard
3	fine	cream	entre	thank	wish	live	name	includ	includ	plate	onion	mayb	probabl	close	indian	leav	decor	like	final	includ	delici	fantast	bring	fantast	tast
4	surpris	dessert	reserv	chees	make	decent	instead	select	wine	delici	fill	fine	famili	hope	lamb	return	like	tast	leav	probabl	return	high	appet	ambianc	impress
5	plate	butter	pasta	strip	probabl	onion	especi	pretti	like	tast	fine	final	includ	wish	rice	finish	fine	onion	light	make	regular	wonder	fast	decor	final
6	impress	hope	wonder	leav	tasti	tasti	probabl	plate	bread	beef	high	brought	bite	butter	spice	person	brought	green	qualiti	person	person	fresh	chang	wish	especi
7	high	crispi	style	onion	awesom	brought	like	fresh	extrem	fill	leav	hard	overal	hard	thank	impress	leav	water	fine	surpris	overal	flavor	especi	light	wrong
8	thank	live	dessert	live	week	size	fresh	spici	make	flavor	instead	care	pasta	soup	garlic	couldnt	half	prepar	butter	couldnt	mayb	awesom	finish	town	lamb
9	pretti	fast	chees	cream	hous	free	hand	indian	outsid	fish	return	couldnt	fine	stop	decent	prepar	chang	bite	main	leav	extra	decor	thank	close	authent
10	hand	soup	butter	includ	pasta	name	stop	sweet	wife	stop	qualiti	return	bring	manag	entre	brought	steak	wrong	onion	thank	make	help	total	appet	strip

For k=200

The results indicate the various topics present in the dataset. As shown in the comment, each topic belongs to a specific category in all of the cases. As reviews are common except the cuisine, I see some of the common words in each of the topics which may not belong to specific cuisine/category. LDA does a good job in identifying the terms belonging to specific category i.e. cuisine. For e.g., Pizza and Pasta for Italian cuisine; Taco, Burrito for Mexican cuisine and so on. As I increased the value for k, I observed that some words were part of few topics instead of one. Also, the topics with general terms were increased. Moreover, we can see that as we use more records we can get accurate results. For less records terms can be repetitive but when you are using more record it will be easier to label the class according to the topic. As our sample dataset is for food reviews, we can see many common terms such as quality, atmosphere, options, services, etc in each of the 5 topics. So, when we executed LDA with k=6 and 10, the common terms were identified and put together as different topics along with original 5 category topics. When K further increased to 50,100 and 200, we observed resulting topics started containing more common terms and some terms were part of multiple topics.

Here are some sample reviews which we have in review.json file and extracted.

“I ordered the Tandoori Chicken and also the garlic rice and garlic bread with a side order of gulag. Everything was really good. “

“The service started off super attentive. My glass would reach half full and it would be filled again before I noticed. By the time we were winding down the service disappeared unless we waved, called and made efforts to squire attention. “

Word2Vec:

I train the text file of yelp dataset of the cuisine that we choose and then I try to search words and figure out top 20 words which are similar with that words. I try it for different words and I give 10 words from that 20 words. The results are as below:

Search_Words= taco, quesadillas, tamales										
	1	2	3	4	5	6	7	8	9	10
word	chimichangas	tortas	sopes	burritos	pastas	quiches	nachos	eggrolls	chimis	fajitas
Sreach_Words = Chinese, French , Mexican										
	1	2	3	4	5	6	7	8	9	10
word	Italian	Indian	European	Latin	Punjabi	American	quebecois	Tibetan	Caribbean	Belgian
Sreach_Words= Water, drink, wine										
	1	2	3	4	5	6	7	8	9	10
word	beer	taquila	wine	pinot	cocktail	sauv	malbec	mimosa	noir	cabernet
Sreach_Words= pizza , pasta, risotto										
	1	2	3	4	5	6	7	8	9	10
word	seabass	linguine	gnocchi	seafood	porcini	gamberi	orzo	ravioli	trout	fettucini

For the search words “tacos, quesadillas, tamales” We can see here, word2vec did a good job in identifying the different type of dishes from different cuisine, as we get the dishes from Mexican, French, Indian, here I show only some dishes from the result. when we search for the words “wine, water, drink” it is giving perfect output as different type of wines, beer, cocktails.

When I search for the words “Chinese, French, Mexican” it gives me output of words like Italian, Indian, Tibetan and many more which are different type of cuisine like the search words so word2vec did a great job. When I search for words like “pizza, pasta, risotto” then I got the dishes like seabass, trout, porcini which all are Italian dishes so again I got the great results of Italian dishes when I give input of Italian dishes.

Word2vec does a good job in identifying words used in similar context i.e. Indian cuisine - Mexican cuisine, drink wine – drink beer, etc. In contrast to LDA, word2vec doesn’t provide any words which belong to the same category as an input word. This can be observed in the first example given above.

The main result which I got is as below from which I give above result of 10 words. It is one of the 4 results.

```
> ana=word_analogy(file_name =
+                    "E://ADM//vec.bin",search_words = "water drink wine",num = 20)

Word: water  Position in vocabulary: 313

Word: drink  Position in vocabulary: 283

Word: wine  Position in vocabulary: 203
> ana
         word              dist
1        beer 0.647433698177338
2     tequila 0.581628203392029
3       wines 0.579143941402435
4       pinot 0.566227197647095
5    cocktail 0.563498854637146
6   margarita  0.55925190448761
7    beverage 0.551037132740021
8        sauv 0.542467713356018
9     chianti 0.537487208843231
10     malbec  0.53602808713913
11    sangria 0.510560214519501
12  sauvignon 0.507918536663055
13   cabernet 0.501289904117584
14     mimosa 0.499555766582489
15       noir 0.497622787952423
16 sangiovese  0.49484446644783
17 chardonnay 0.494783669710159
18  cocktails 0.493963032960892
19  champagne 0.490682899951935
20     shiraz 0.483722060918808
```

Similarly I got results for the other search word

Enron Email dataset:

Sometimes it becomes tuff to give labels to the topics when your dataset does not contain any and we don't have information about that. Same thing happens in **Enron Dataset** I run LDA on the Enron dataset for high records and here is the output for the k=5 and top 10 words. First, I extract data from Enron dataset for which I do python code and extract every email id and store it in different text files.

	1	2	3	4	5	6	7	8	9	10	Comments
Topic 1	pleas	content	thank	folder	date	attach	messag	text	type	version	**Attachments**
Topic 2	power	energi	california	price	electr	util	rate	generat	market	cost	**Business**
Topic 3	enron	http	mail	jeff	john	mark	corp	creat	time	steve	**Employees**
Topic 4	compani	market	trade	servic	busi	deal	manag	report	project	time	**Trading**
Topic 5	meet	call	inform	discuss	week	question	issu	follow	request	schedul	**Meetings**

For k=5

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
1	shuck	enron	divulg	percent	consum	encod	insignif	enron	kansa	passag
2	trata	will	paltri	first	plea	realli	clpearc	bodi	seco	smud
3	calvert	said	gracious	requir	hear	meter	cartwheel	electr	skippingston	castro
4	commit	energi	lebanon	bush	past	major	improvi	power	blay	midpoint
5	divest	power	enlighten	chang	util	recommend	previous	state	blith	eral
6	footbal	compani	chene	corp	less	execut	sherriff	said	bundesbank	fledgl
7	heineman	price	clapp	support	west	idea	coal	rate	disp	mccoppin
8	nanotechnolog	california	elind	need	becom	america	pdowd	year	ebazar	serc
9	piotrowski	state	gdreed	document	user	cheney	peacekeep	will	evolutionari	alberta
10	repercuss	util	kris	power	hold	panel	undercollect	energi	kawamoto	benson

For k=10

Here I don't know any label but still I labelled the data according to the words which appear in this. SO, according to me I categorized the data like above pic where I imagine that this can be a label of that topic.

The figure shows the results I received for K=5. As there were not a predefined category for this dataset it was hard to label each topic. The topic 3 contains terms like Jeff, John, and mark so we can say that this topic represents the employees of the company. With terms meet, call, schedule in topic 5, it can be labelled as Meetings. similarly, other topics can also be labelled as business, attachments, and trading. Results were somewhat similar when I provided value for K other than 5.

Here is the sample example of the text file of the email dataset in which the emails were extracted. I have some snippets by which we can have idea about the conversation between Enron employees through emails.

"Here I don't know any label but still I labelled the data according to the words which appear in this. SO, according to me I categorized the data like above pic where I imagine that this can be a label of that topic."

"Jeff The toad doesn t know how to please your wife Here is my friend Marc s e mail marc sondheimer quokka com When you pull a squirrel out of the water try and sell him bandwith Sean"

"Thanks I ll soften the expropriation language I don t think they can disclose the 1 million on the product liability case since the entire package appears to be one negotation all the pieces of which likely have to be agreed to or no settlement But you re point s well taken and I ll try to acount for it in the answer Finally I can t see disclosing the fact that I m infringing on someone s patent in my annual report Seems like I d have a hard time defending myself in court if they sued me if I d admitted it to it in my annual report "

Word2Vec:

I train the text file of Enron email dataset and then I try to search words and figure out top 20 words which are similar with that words. I try it for different words and I give 10 words from that 20 words. The results are as below:

Search_Words= cost, steve, time										
	1	2	3	4	5	6	7	8	9	10
word	Umbrella	potential	McConnel	Bill	Linda	planning	Rs	Bluesteins	853	decided
Sreach_Words = Compani, servic, busi										
	1	2	3	4	5	6	7	8	9	10
word	velvety	Terry	Messaging	Gwen	Walls	Marta	jdoyle	Rudnick	Cameron	gacynth
Sreach_Words= California, John, Schedul										
	1	2	3	4	5	6	7	8	9	10
word	Austrlia	Tee	Friday	Britt	Nigeria	Box	Mon	sucker	osed	Cowen
Sreach_Words= deal, follow, date										
	1	2	3	4	5	6	7	8	9	10
word	Assignment	Componer	RDU	Dasovich	August	Washignton	minute	build	spokesman	Susan

When I am searching for words “cost, steve, time” it gives me output like Bill, RS, 853, ,minute which terms are related . For example, RS and 853 is related to cost similarly Bill is related to John at last minute is related to time. So word2vec is giving proper output. When I am searching for words “company , service, busi” I get results related to this three words.

When I search for words “California, John, Schedule” then I get the output which is related to this words for example Austrlia and Nigeria are related to California, Britt is related to John, Mon and Friday are related to schedule, so we got the perfect results .When I am searching the words “deal, follow,date” it gives me term related to these words as the output. Which is accurate as deal is connected with assignment and build, date is connected to August, minute.

The main results from which I got the above-mentioned result is as below:

```
> ana=word_analogy(file_name =
+                    "E://vec.bin",search_words = "california john schedul",num = 20)

Word: california  Position in vocabulary: 10754

Word: john  Position in vocabulary: 1441

Word: schedul  Position in vocabulary: 27080
> ana
        word              dist
1        Tee 0.627698302268982
2     sucker 0.627017557621002
3   websites 0.625972807407379
4  Australia 0.399339914321899
5    Nigeria 0.385163158178329
6        BOX 0.371312588453293
7      Britt 0.370872139930725
8        MON 0.370653182268143
9    hpl0906 0.370445102453232
10    myself 0.362861573696136
11      osed 0.359484225511551
12 previousl 0.359216421842575
13    FRIDAY 0.357041954994202
14    AZURIX 0.355440855026245
15     Cowen  0.35517156124115
16    slides 0.344117790460587
17  specials 0.342872977256775
18    MIDDLE 0.341387182474136
19     solid 0.340614289045334
20 Woodlands  0.34009575843811
```

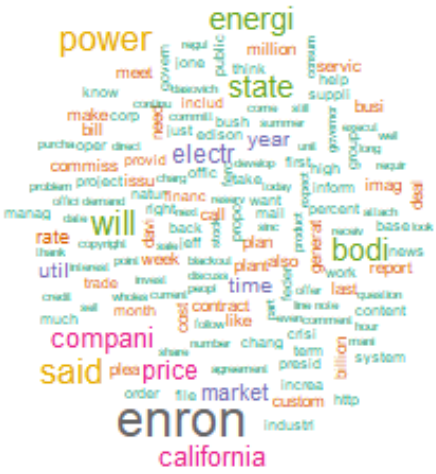
Similarly, I got results for the other search words.

Wordcloud:

I also plot Wordcloud for both of the dataset so that we can see the terms which appears in the dataset. So the wordcloud for both Enron email and yelp dataset is as below:

For Yelp dataset:

For Enron Email dataset:




```
freq <- sort(colSums(dms), decreasing = TRUE)

dark2 <- brewer.pal(8, "Dark2")

wordcloud(names(freq), freq, max.words=100, rot.per=0.2,colors=dark2)
```

VII) Analysis/Learning/Challenges:

-The dataset used for the analysis was yelp dataset and Enron Email dataset which is huge and with different characteristics. Firstly the data is pre-processed and stored into another files.

- Pruning words by frequency i.e. removing words which occur less frequently is a better way to prepare the data for analysis. This will reduce the size of data and will make it easier for computation.

- LDA did perform well on the dataset and represented topics in a fairly better way. Although word2vec model does not provide topics, we can search for the different words and get similar words as output.

- During the entire procedure, upon comparing the influence of simplest of procedures like pruning, labeling, tf-idf weighting, formation of document-term matrix etc. provided a deeper understanding of the concepts. And to label the data which does not contain any label is also tuff.

- Yelp dataset and Enron email dataset are very huge dataset, so loading entire dataset, converting it to .csv format and reducing the data took time and effort.

- Once dataset was obtained, due to its large volume it became too tuff to train the models and get the result as system keeps crashing.

- The main Problem which I get in this project is that to give labels to the terms which is not even label values. As I am using “word2vec” and “LDA”. And this both methods are working properly as we do not have any error. It will get slow only if you using the whole file together as it has so many class.

-It slows down your processor when you increase size of the input files. And it is not working well because of the system configuration if you have highly configured system then this problem will not arise.

XI) Conclusion:

- The LDA learns a document vector that predicts words inside of that document and cluster words that tend to co-occur together into interpretable topics.

- This can be seen in the results we got for Yelp dataset and can be verified with the sub category of cuisine types chosen.

- In word2vec by training the model on a large corpus, word embeddings with encoded semantic information can be obtained

- The word2vec model learns a word vector that predicts context words across different documents.

- Based on our experiment results we can say that LDA is more suitable for Sentiment analysis and word2vec is for Semantic analysis