Topic modeling on Yelp Database

Rojina Deuja
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, USA
rojinadeuja33g@gmail.com

***Abstract*—In this study, we carry out topic modeling to discover the most talked about topics in a discussion platform. The platform used is the Yelp website, where we extract the user reviews and then use Latent Dirichlet Allocation (LDA) to discover the set of most popular topics. The discovered topics can be utilized to derive essential business knowledge that can fuel business decisions as per user activity. We further implement a more powerful- Word2Vec model in order to validate the results that we have obtained. With the help of these two model, we create and optimize the process of Topic Discovery in large datasets.**

***Index Terms*—topic modeling, clustering, text mining, natural language processing**

I. Introduction

Customer reviews not only have the power to influence consumer decisions but also can impact a company's credibility. Reviews have been successfully utilized to analyse user activity, customer opinions and sentiments, popularity of products and even to forecast sales. A popular use of review data is to discover popular topics among the user of a business through Topic Modeling. It originated from latent semantic indexing [1]. Topic Modeling is an unsupervised machine learning technique that processes a set of documents, detects words/phrases inside them and then clusters related words together in groups. These groups are what form a topic. They can be efficiently used to automatically discover topics in a large set of documents, that may not be apparent otherwise.

II. Research Question

C2C(Customer-to-Customer) is a business model where a company provides a platform of communication for sales between sellers and buyers. My motivation for choosing this area is to study the role of the intermediary businesses that operate between them. The question I want to answer is: How to discover the relevant topics in a large set of unstructured documents ?

III. Project Description

In this project, we analyze the Yelp data set. The reviews in the data set carry critical information about customer interests. If we are able to extract the important topics from all the discussions that goes on between the users in the website, it can be utilized to make better business decisions. For this purpose, we want to carry out Topic Modelling on the Yelp data set. This is interesting particularly because, from the thousands of data about various businesses, it can help us discover what things are of high value to the consumers. The diagram in Fig. 1 shows the E-R analysis workflow of our project. We have five components named as $business$, $review$, $user$, $checkin$ and tip. Each component represents a table with it's set of attributes to describe that component. There is One-to-Many relationship between $user$ and $review$, tip. Similarly, there is a Many-to-One relationship between $review$, tip and $business$. There is also a One-to-Many relationship between $business$ and $checkin$.

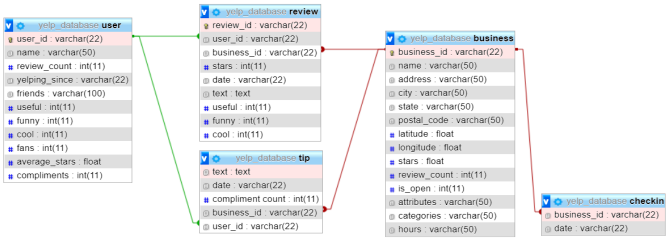

Fig. 1. E-R data analysis workflow

IV. Data set

The data set was published by Yelp, which is a business directory service and crowd-sourced review forum. It contains 6,685,900 reviews of 192,609 different businesses, given by 1,637,138 users.

The data set is a collection of six JSON files namely: $business$, $review$, $user$, $checkin$ and tip. The attributes are a combination of numeric and text data. For numeric data, we have discrete attributes like ids and continuous attributes like date, stars etc. The data is formatted in a structured manner. All the respective records are divided into relational attributes and the samples are categorized into individual category. Source: [Link].

V. Data modeling

The data provided by Yelp is large and structured. Although the data is relational, we are only interested in a handful of these features for our study. The dataset containing records about the $business$ and the $reviews$ are focal to our study. Since a large portion of the data involves text mining of the reviews, we store the data in a No-SQL format.

A. Conceptual modeling

For designing the conceptual database model, we considered the following:

1) Business requirement: First we ask the question: What is the requirement of the business? The Yelp website is concerned with getting reviews for businesses from the users. In this study, we process the reviews to extract the most popular topics. Users, Businesses, Reviews etc are distinct elements of our database which consists of their own set of attributes. Thus, we divide our records into five entities, each of which represents a distinct table in our model. Maintaining the records in different tables helps keep related information together and at the same time joins can be used to access any interconnected data easily.
2) Scope: Next we answer the questions related to the scope of our study. For example: Can a review exist without a business? Can it exist without a user? Can a business have multiple reviews? and so on. These relationships can be defined in terms of indexing the tables that allows us to create foreign-key relations between tables.
3) E-R Diagram: After deciding on the structure of our model, we represent it in the terms of an E-R diagram as shown in Fig 1. The database has five tables: $business$, $review$, $user$, $checkin$ and tip, each with some relationship with another table. These relationships can be One-to-One, One-to-Many or Many-to-One as denoted by in the diagram.

B. Logical Modeling

For the logical data modeling, we consider how different entities are identified in our database. For example: Each business is uniquely identified by the $business\_id$ field. The $business\_id$ is also present in each review to mark which business is being identified by the user. We specify five tables for the five entities identified in the Conceptual modeling stage. We set primary and foreign key specifications in the tables. We also normalize the data up to 3rd Normal Form. The purpose of normalization to remove any redundancy in the database and to provide stability. This saves space and time used in accessing the database. In addition to this, the design becomes much more elegant and easy to comprehend. This will be a baseline for our physical data model.

C. Physical Modeling

Physical Modeling is the process of capturing the details of the technical implementation. For the scope of this study, we are only interested in the reviews associated with each business. Since $review$ and $business$ are frequently queries together, we embed the reviews inside $business$ has a separated field- $reviews$. This helps us achieve better performance since no joins are required between $review$ and $business$. As for the other entities like $user$, tip and $checkin$, we simply store is using References.

We created scripts to store the respective data as collections in our database. For instance, data associated with business was stored in the collection named $'business'$. We also created scripts to embed the review data into the collection $'business'$. For this study, we are only concerned with business falling under the category 'Restaurants' so, we created scripts that removes all businesses that are not restaurants. For further data pre-processing, we also removed businesses with review count less than 10 and only retained the most recent reviews i.e. reviews from the year 2018. Finally, we also removed any business that did not have a review and stored it in a collection named $'business_restaurant'$. Once all the scripts were ready, the MongoDB database was created with the name $yelp\_database$ and the scripts were run using PyMongo. The database would have an additional collection named $'topic'$ which holds the results from the LDA Topic Modeling.

VI. Data Modeling for Analysis

The reviews data is in a text format. We will perform Topic modeling to discover hidden patterns lying in the discussions within the reviews. Clustering similar topics together can help identify the significant topics of discussion among users that can help improve the business further.

A. Topic modeling using LDA

Latent Dirichlet Allocation (LDA) is a probabilistic topic model introduced in [2] which is used in Natural Language Processing (NLP). It is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, it assumes that each document is a mixture of a small number of topics and that each word in the document attributes to one of the topics in that document. We use $genism$'s $ldamodel$ for implementing our research .This module allows both LDA model estimation from a training corpus and inference of topic distribution on new, unseen documents..For our LDAmodel, we set $alpha$ = $'auto'$ and eta = $'auto'$, which means that we are automatically learning two parameters in the model that we usually would have to specify explicitly.

a) Optimal Number of Topics: Topic Coherence is a measure that can be used to evaluate topic models.Topic Coherence is defined as the average of the pairwise word-similarity scores of the words in the topic. Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that can be interpreted semantically and topics that are mere artifacts of statistical inference.

A good model will generate coherent topics, i.e., topics with high topic coherence scores. Good topics are topics that can be described by a short label, therefore this is what the topic coherence measure should capture. For our model, we choose the number of topics K as 9. This decision is supported by running a hyper-parameter tuning algorithm using various value of K on our LDA model as shown in Fig. 2 , the highest Coherence was achieved when K=9.

	Topics	Coherence
0	8	0.325261
1	9	0.342279
2	10	0.333259
3	11	0.314275
4	12	0.332904
5	13	0.323092
6	14	0.334951

Fig. 2. Topic Coherence for different values of K

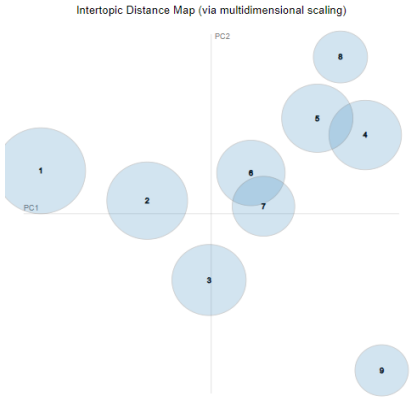

Fig. 3. Intertopic Distance Map

b) Interpretation of Results: From the LDA model, we get a set of $K = 9$ unique topics and a list of words (terms) that belong to the topic as shown in Fig. 4. Each word also contains the probability of falling into the respective topic. Topics learned from a statistical topic model are formally a multinomial distribution over words, and are often displayed by printing the 10 most probable words in the topic. These top-10 words usually provide sufficient information to determine the subject area and interpretation of a topic, and distinguish one topic from another. [3]

	Topic # 01	Topic # 02	Topic # 03	Topic # 04	Topic # 05	Topic # 06	Topic # 07	Topic # 08	Topic # 09
0	highly_recommend	soup	pizza	coffee	sushi	taco	told	burger	happy_hour
1	wine	noodle	wing	cream	roll	salsa	manager	breakfast	beer
2	dining_room	pork	crust	spring_roll	bowl	burrito	waitress	bacon	sandwich
3	room	shrimp	italian	cake	spicy	chip	left	egg	game
4	dining	thai	pasta	chocolate	ramen	mexican	business	drive_thru	local
5	gluten_free	mashed_potato	topping	dessert	fish	bean	waiting	sandwich	year
6	dessert	broth	slice	milk	soup	carne_asada	owner	brunch	pulled_pork
7	steak	pork_belly	garlic	cafe	buffet	chip_salsa	walked	onion	five_star
8	chef	plate	deep_fried	shop	flavour	black_bean	away	toast	open
9	reservation	cooked	lobster	vegan	korean	tortilla	looked	onion_ring	music

Fig. 4. Topics obtained using LDA

From the table, we manually generate human-interpretable labels for each topic by looking at the terms that appear more in each topic. The labels for the topics are shown in Fig. 5.

B. Word Embedding using Word2Vec

After getting results from the LDA model, we use a more powerful model - Word2Vec to see if our findings are actually consistent. Word2vec is a group of related models that are used

Topic	Topic Label
1	Fine Dining
2	Thai Food
3	Italian Food
4	Bakery
5	Asian Food
6	Mexican Food
7	Customer Experience
8	Fast Food
9	Happy Hour

Fig. 5. Assigned Topic Labels

to produce word embeddings. These methods are prediction based in the sense that they provided probabilities to the words. Word embedding is the collective name for a set of language modeling and feature learning techniques in NLP where words or phrases from the vocabulary are mapped to vectors of real numbers. Word Embeddings are the texts converted into numbers and there may be different numerical representations of the same text. We use $genism$'s $Word2Vec$ class for implementing this model. The $dimension$ of the word vector is set to 140, the $window$ of words that are scanned is set to 2 since the sentences in the reviews weren't too long. We also ignore all words that have a total $count$ less than 100.
We use the Skim-gram model introduced by Mikolov et al. [5], which is an efficient method for deriving vector representations of words from large unstructured data. The Skip-Gram model breaks a sentence into (target, context) pairs. Given the vocabulary size V, the skip-gram model learns word embedding vectors of size N. The model learns to predict one context word (output) using one target word (input) at a time.

a) Interpretation of Results: We evaluate our model by looking at the words that are predicted closer to each other. For example, from the figure, we can see words such as Coffee, Almond, Vanilla are predicted to be closely related to the word 'Milk'. After conducting a couple of such verification, we can say that our model is able to embed the words well.

b) LDA vs Word2Vec: A hybrid model combining LDA and Wordd2Vec has been used for document feature extraction [**?**]. Once we get the results from both the models, we compare them to determine how LDA fares as compared to the Word2Vec. We plot similar words under each 3 topics from the LDA model using t-SNE plot as shown in Fig. 7. Then we verify if the words under the LDA topics are actually clustered together by the Word2Vec model. This integrates the contextual relationships among words and allows us to study how the clustering is influenced by statistical vs contextual sampling.

VII. Data Modeling for Web-Based Visualization

The data used for the Topic Modelling had a dimension of 8688 rows and a single column that contained the reviews. After the topic modelling is carried out, we extract the 9 topics and their 10 top words associated with that topic. For the purpose of the visualization, we want to visualize the topics and the words in a Word Cloud. Thus, no dimensionality

```
model.wv.most_similar('milk')
```

```
[('coffee', 0.5586003065109253),
 ('almond', 0.5159417986869812),
 ('vanilla', 0.4771198630332947),
 ('strawberry', 0.4563044309616089),
 ('coconut', 0.45427215099334717),
 ('sugar', 0.45378971099853516),
 ('matcha', 0.4314887523651123),
 ('tea', 0.42589133977890015),
 ('latte', 0.4252324104309082),
 ('bubble', 0.42409974336624146)]
```

Fig. 6. Words similar to 'Milk'

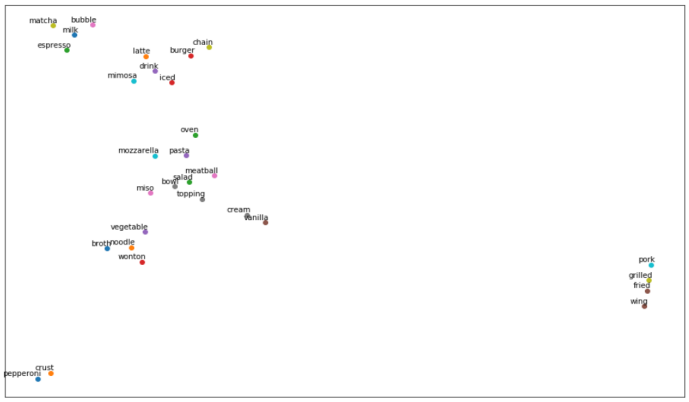

Fig. 7. Subset of Vocabulary showing similar words togther

reduction techniques were required. We use the $d3.js$ and $d3 - react - cloud$ to create the visualizations.

The first visualization that we create is are the word clouds for all the 9 topics. We used $d3 - react - clould$ for creating the Word Clouds. Each term in the cloud is represented by a random color, size and rotation. If the user clicks on any term, the probability of the term falling under the topic is shown. The Word Cloud for Topic 0, i.e. Fine Dining is shown in the Fig. 8.

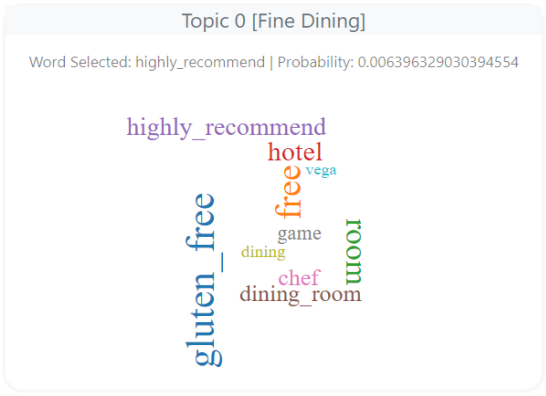

Fig. 8. Topics visualization using Word Cloud

The second visualization that we create using $D3.js$ is a horizontal bar chart shown in Fig. 9 that plots the average topic coherence for each topic. Average topic coherence is the sum of topic coherences of all topics, divided by the number of topics. The higher the coherence values, the better the predictions. Each bar in the bar chart changes color on mouse over. The bar chart is sorted by increasing values of coherence. Each bar has the respective topic label alongside the y-axis.

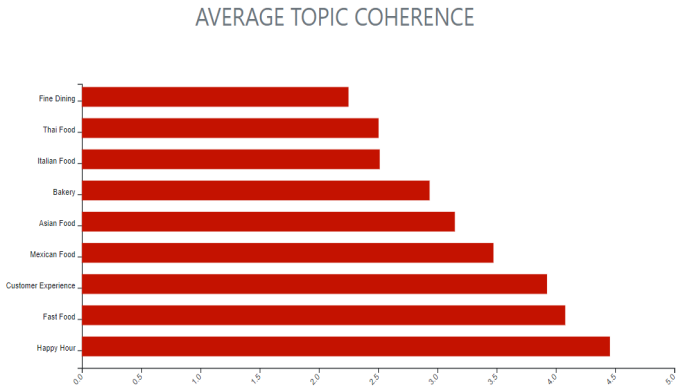

Fig. 9. Row Chart showing Topics Coherence

VIII. Conclusion and Future Work

In this study, we worked on discovering topics in the dataset. We studied how LDA and Word2Vec models work on textual data and how they can be optimized. We also realized the importance of data pre-processing to filter out less meaningful information and to improve our model. While text mining and natural processing was a very interesting area to work on, more can be done. After retrieving the topics, we need to think about how they can be used acquiring deeper business knowledge. A possible extension to this could be creating recommendation systems to users based on the type of reviews that they have given in the past.

IX. Acknowledgements

I would like to thank Dr. Mohammad Rasedul Hasan for the valuable guidance and continued support throughout this study. The concepts imparted by him in his lectures has help me understand the fundamentals of both Data Modeling and Machine Learning. His ideas have helped me create better outcomes and also seek a deeper meaning behind the work that I am doing. I am positive that this study will transform into something even better with further supervision from him.

References

[1] Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R, "Indexing by latent semantic analysis", J Am Soc Inf Sci 41(6):391, 1990.

[2] Blei, D.M., Ng, A.Y., Jordan, M.I., "Latent dirichlet allocation" in J. Mach. Learn. Res. 3 (2003) pp. 993–1022

[3] Newman D., Edwin B.V., Wray B. , "Improving Topic Coherence with Regularized Topic Models" in "Advances in Neural Information Processing Systems 24", pp. 496-504, 2011.

[4] Mikolov T., Chen K., Corrado G., and Dean J, "Efficient estimation of word representations in vector space," ICLR Workshop, 2013

[5] Z. Wang, L. Ma and Y. Zhang, "A Hybrid Document Feature Extraction Method Using Latent Dirichlet Allocation and Word2Vec," 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), Changsha, 2016, pp. 98-103.