# Predicting Yelp Review Upvotes by Mining Underlying Topics

**Shuyan Wang**

Quantitative Methods in the Social Sciences
Graduate School of Arts and Sciences
Columbia University

Advisor: John Paisley
Department of Electrical Engineering, Columbia University

Jan 2015

This paper was completed in part-fulfillment of QMSS G5999: Master's Thesis. I am indebted to John Paisley for invaluable advice given. I am also grateful to Rongyao Huang, Sonia Lee, Greg Werbin, Jonah Gabry, Professor Sung-Woo Cho, Beth Kopko, all of whom provided substantive feedback.

**Abstract:**

The paper attempts to predict Yelp review upvotes based on the text content by presenting a model that digs into the correlation between populat topic number a review covers and the “useful” upvotes it gets. LDA_VEM model is applied to extract the subtopics of reviews and mine the most popular topics on Yelp. The paper also looks into other question like the seasonality / trend of Yelp review popularity. In addition to that, the paper compared the frequent words and topics clustered by LDA to vonfirm the validity of LDA mode

## 1. Introduction

As another hyper-successful social network website besides Facebook and Twitter, Yelp has collected millions of reviews of all types of businesses one can think of in the past decade, and has fundamentally changed the landscape of the way people choose and rate service businesses, as well as the way service businesses run themselves. Despite the legend of this billion-dollar worth start-up, there is one issue that is already troubling Yelp and may jeopardize its future thriving: When a restaurant has already had a significant amount of reviews, plus some of them have gained certain level of popularity (measured by the anonymous funny/ useful/ cool upvotes it has received), a new user's review will be buried deep and overlooked the second it comes out. With new users being aware of this, it will certainly impair users' participation, and slow down the metabolism of the network in the long run. Is there a method that we can recognize and choose potentially popular reviews (perhaps those that are informative, fun to read, and perhaps) based on its text content, and automatically "pop it up" to the surface? A good system that is able to recommend high quality reviews can greatly boost user engagement.

The problem above leads us to a very intriguing machine learning question that has obvious significance in reality. McAuley and Leskovoc (2014) brought up a theory in their paper that won the Yelp review challenge that reviews whose text best explains the hidden factors (known as underlying reviews) are more considered 'useful' by human annotators, although most of their paper was on mixed rating of yelp stars and text reviews. This paper will try to address this assumption by summarizing sub-topics from review content and modeling them with popularity, and check the correlation of these factors.

## 2. Relevant Works

Natural Language Processing has made considerate progress in the past years. To understand a certain amount of discrete words and terms, a statistic known as Term Frequency-Inverse Document Frequency was created in 1983 to reflect how important a word is to a document (Salton and McGill, 1983). While the tf-idf reduction has some appealing features, the approach still provides a relatively small amount of reduction in description length and shows little in the way of inter- document statistical structure. To address this flaw, Hofmann(1999) came up with aspect model that structures each word in a document as a sample from a mixture model, which was a great leap forward, and will be described in detail in the modeling part of this paper. However, this model is still incomplete at the level of documents.

The Latent Dirichlet Allocation (LDA) is a hierarchical probabilistic model used to decompose a collection of documents into its salient topics, where for LDA, a topic is a probability distribution over a specific vocabulary. LDA was a considerable success in topic modeling and graphic modeling area as it enables researchers to quickly summarize, explore, and search massive document collections applying machine learning model (Blei et al., 2003). Its goal is to find short descriptions of the members of a collection that enable efficient processing of large collections while preserving the essential statistical relationships that are useful for basic tasks such as classification, novelty detection, summarization, and similarity and relevance judgments. Having been extended to many other kinds of area, i.e., computer vision problems, sound features, survey responses and genetic makers, LDA will be applied in social network data in my research to find structure in texts and exploit grouped data (Blei, 2012).

LDA has gained much popularity in the past decade, applied ranging from topic modeling to computer vision. Interesting research papers include LDA for Tag Recommendation (Krestel et al., 2009), and application in human action recognition (Wang et al., 2007). The former used LDA in tagging system, showed that LDA-based approach is able to elicit a shared topical structure from the collaborative tagging effort of multiple users, whereas association rules are more focused on simple terminology expansion. The latter improved action recognition based on motion words by using hierarchical probabilistic model. Compared to recommendation system and document summarization, LDA has been relatively new to social network media, and it is interesting to explore this area.

There is a long line of works studying social network texts using varied topic models including K-means clustering, hierarchical topic model (hLDA), query related Supervision knowledge (ShLDA), duplicate Relation constraints (RShLDA) etc. Huang, Rogers and Joo (2014) used online Latent Dirichlet Allocation algorithm on Yelp dataset to point out demand of customers, present breakdown of hidden topics and predict stars per hidden topics discovered. An integrated model of RShLDA and ShLDA was applied to exploit the topic hierarchy for online video organizing by utilizing associated metadata to cluster the returned results into semantic groups according to its involved subtopics (Sang & Xu, 2012). There also have been successful clustering of terms and text documents using non-negative matrix factorization techniques, such as factoring of 90,000 terms in emails to 50 clusters (Berry and Browne, 2005). Other than useful upvotes, other measure of review quality such as trustiness, reliability, agreement and honesty was also brought up to detect Amazon spam reviews using graph-based

algorithm (Shinzaki et al, 2013). To cope with reality, Partially Labeled Dirichlet Allocation (PLDA) besides supervised and unsupervised learning is presented to make use of the unsupervised learning machinery of topic models to discover the hidden topics within each label, as well as unlabeled, corpus-wide latent topics (Ramage et al, 2011).

## 3. Methods

The usefulness of a review, denoted by $U_d$, is the useful upvotes a review *d* posted at time *t* is supposed to get based on its content on the current time point. In this model, we assume that $\tilde{U}_d$, the actual upvotes is an unbiased estimate of the usefulness. The hypothesis is that there are three features of a review that will influence its usefulness in human's eyes: the number of latent topics its best describes, the rating it gives, and the amount of time since it has been written.

The rating deviated from average rating is also considered a factor, as the paper assumes that the a more "extreme" rating would be considered more useful than an opinion that seems to give little "attitude". That is to say, people would be more willing to see either a five-star or one-star rating than a three-star rating. This is no more than a guess that this paper would like to test out with data.

This model also has a time-series factor, as the longer a review is online, the more views it would have and surely, there is a better chance it would get more useful upvotes (if it truly is useful). Meanwhile, this is also a hypothesis to be tested that on what time scale would the upvote number become significantly different (and if it is really different). It may also be relevant to the amount of active users on Yelp, and we will look further into it in the following data cleaning part.

(1) $$U_d = \theta_i * \Sigma + \beta_1 * l_d + \beta_2 * (r_d - r_{mean}) + \beta_2 * (t_{now} - t) + \varepsilon$$

*Latent Dirichilet Allocation*

LDA associates each document *d* is an element of *D* with a *K*-dimensional topic distribution $\theta_d$ (i.e., a stochastic vector), which encodes the fraction of words in *d* that discuss each topic. That is, words in the document *d* discuss topic *k* with probability $\theta_{d,k}$.

Each topic *k* has an associated word distribution $\phi_k$, which encodes the probability that a particular word is used for that topic. Finally, the topic distributions themselves ($\theta_d$) are assumed to be drawn from a Dirichlet distribution.

The final model includes word distributions for each topic $\phi_k$, topic distributions for each document $\theta_d$, and topic assignments for each word $z_{d,j}$ . Parameters $\Phi = \{\theta, \phi\}$ and topic assignments *z* are traditionally updated via Gibbs sampling. The likelihood of a particular corpus *T* (given the word distribution and topic assignments for each word) is then

(2) $$p(T|\theta, \phi, z) = \prod_{d \in T} \prod_{j=1}^{N_d} \theta_{z_{d,j}} \phi_{z_{d,j}, \omega_{d,j}}$$

where we are multiplying over all documents in the corpus, and all words in each document. The two terms in the product are the likelihood of seeing these particular topics ($\theta_{z_{d,j}}$), and the likelihood of seeing these particular words for this topic ($\phi_{z_{d,j}, \omega_{d,j}}$). Below (Table 1) is a summary of the notations this paper will use for models.

**Table 1. Notations of the Model**

| Symbol | Description |
|---|---|
| $t$ | time (date) $t$ a review $d$ is posted |
| $d_t$ | review $d$ posted at time $t$ |
| $\tilde{U}_d$ | usefulness (measured by 'useful' upvotes) of a review $d$ |
| $\theta_i$ | $K$-dimensional topic distribution for item $i$ |
| $\phi_k$ | word distribution for topic $k$ |
| $K$ | Number of latent dimensions/topics |
| $\omega_{u,i,j}$ | $j^{th}$ word of user $u$'s review of item $i$ |
| $Z_{u,i,j}$ | topic for the $j^{th}$ word of user $u$'s review of item $i$ |
| $l_d$ | length of a review $d$ |
| $r_d$ | rating that a review $d$ gives to item $i$ |

## 4. Dataset and Results

### 4.1. Overview of the Dataset

The dataset used in this paper is the Yelp Academic Dataset. The *Yelp Academic Dataset Challenge* Data is constituted of five parts in the format of JSON(JavaScript Object Notation) file: business data, checkin data, review data, tip data, and user data. It covers massive data from Phoenix AZ, Las Vegas NV, Madison WI, Waterloo ON, and Edinburgh UK of 42,153 businesses, 320,002 business attributes, 31,617 check-in sets, 252,898 users, 955,999 edge social graph, 403,210 tips, and 1,125,458 reviews. Below

(Table 2) is a summary of the crucial statistics that can give the readers a high-level peek into the dataset. And this paper would mainly focus on the review data that contains ten variables itself and business data.

**Table 2. Basic Statistics and Structure of Yelp Dataset**

| Data | Number of records | Missing Data | Information Categories |
|---|---|---|---|
| **Business** | 42,153 | No | Business id, Full address, Hours, Open, Categories, City, Review Count, Name, Neighborhoods, Longitude, State, Stars, Latitude, Attributes, Ambience |
| **Check-in** | 31,617 | No | Check-in information, Type, Business id |
| **Review** | 1,125,458 | No | Votes (funny, useful, cool), User id, Review id, Stars, Date, Text, Type, Business id |
| **Tip** | 403,210 | No | User id, Text, Business id, Likes, Date, Type |
| **User** | 252,898 | No | Yelping Since, Votes (funny, useful, cool), Review Count, Name, User id, Friends, Fans, Average Stars, Type, Compliments, Elite |

### 4.2. Crucial Statistics of the Review Data and Sampling

The research question is how to predict the popularity of a review based on its text content. Before looking into this question, there are three sub-questions the paper would like to answer: 1. What is the popularity of all the reviews like (such as distribution, mean,

deviation, skewness)? 2. What does the review content looks like (such as length of text, high frequency words, etc.)? 3. What is the effect of the factor time on popularity? (Is it linear, exponential, or non-related?) Information on the popularity statistics can help with model selection and optimization, and the first question is answered by an overview statistics of the review data (Table 3) below.

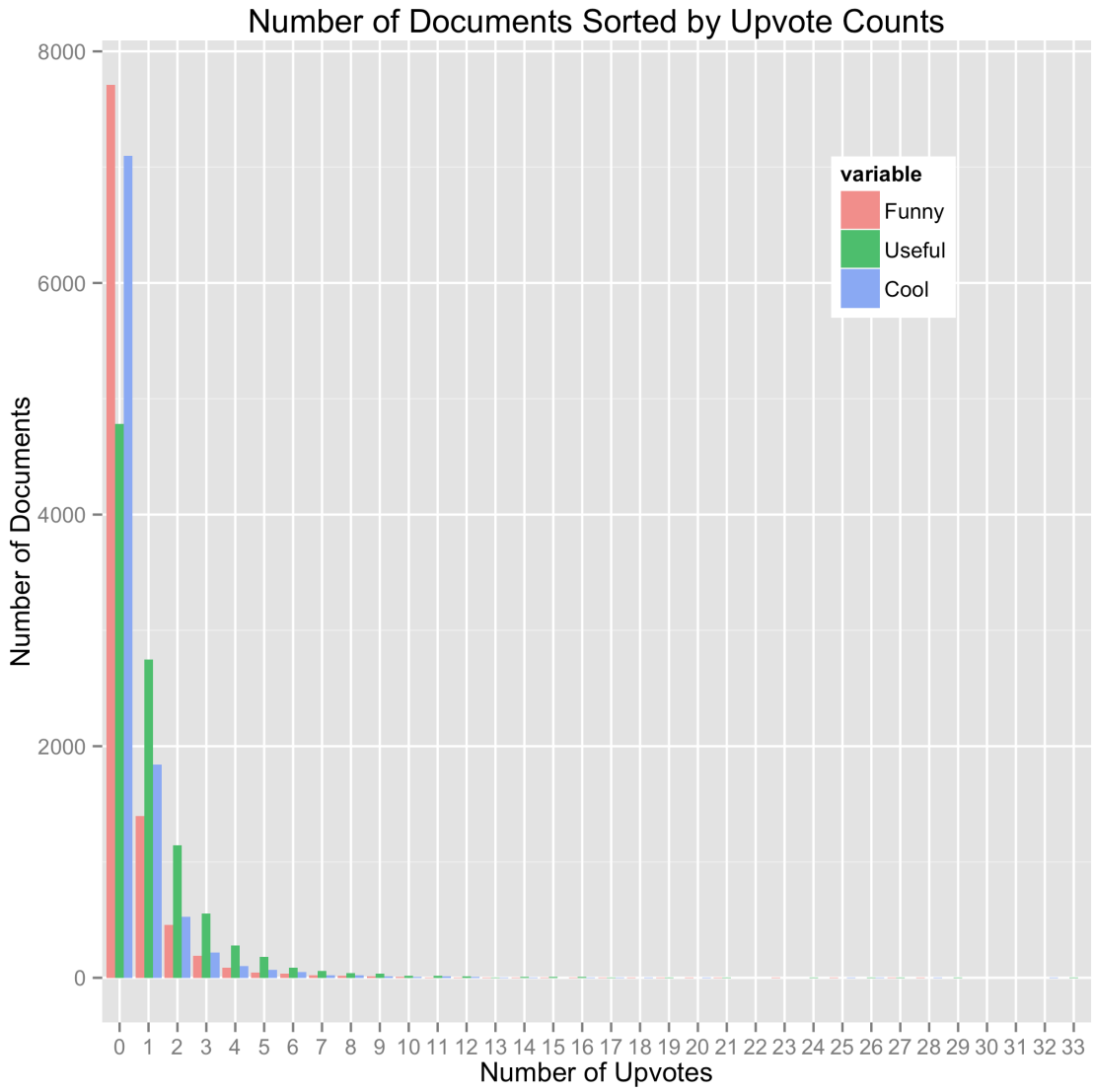
**Table 3. Overview Statistics of the Review Data**

| Variables | Upvotes - funny | Upvotes – useful | Upvotes - cool | Stars | Date (days) | Text length |
|---|---|---|---|---|---|---|
| **Mean** | 0.525 | 1.132 | 0.653 | 3.737 | 4/25/12 | 129 |
| **Median** | 0 | 0 | 0 | 4 | 9/5/12 | 95 |
| **Min** | 0 | 0 | 0 | 1 | 10/18/04 | 1 |
| **Max** | 141 | 166 | 137 | 5 | 7/16/14 | 1034 |
| **Standard Deviation** | 1.634 | 2.125 | 1.712 | 1.299 | 636.293 | 117.52 |

As LDA algorithm creates very large sparse matrix and requires a considerable amount of internal storage to run, this paper will sample 10,000 reviews randomly from all the Yelp reviews created in 2012. The reason the analysis focus on 2012 is that 2012 reasonably far from the current time point and has given reviews plenty of time to be viewed and rated. Meanwhile, in the year of 2012, the website Yelp has come to the growth plateau with a rather stable amount of active users and traffic that guarantee

healthy interaction. This move will influence the assumption of time series factor mentioned earlier, and will be discussed later.

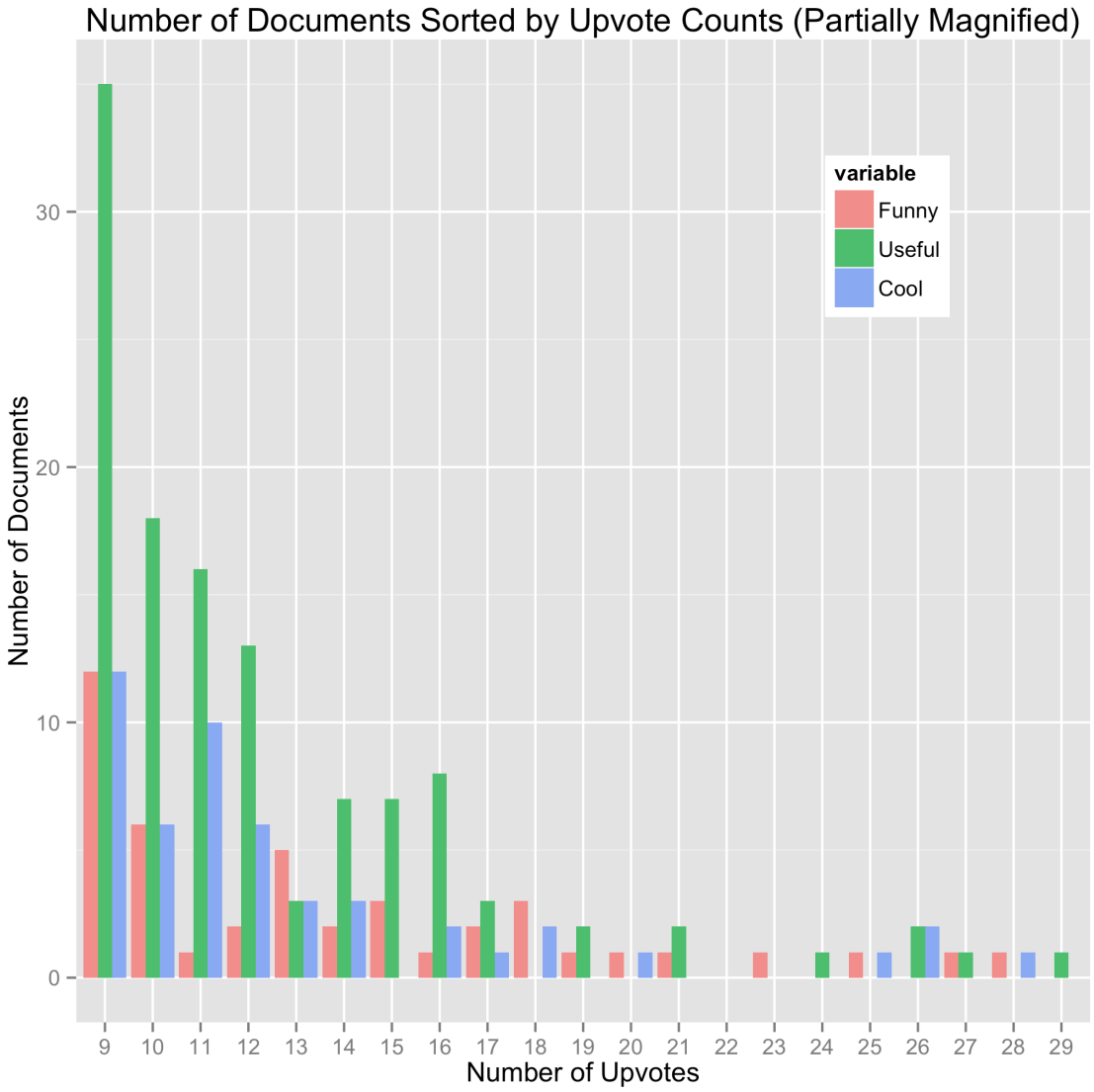
**Figure 1. Number of Documents Sorted by Upvote Counts**




From the bar plot (Figure 1&2), it can be seen that out of 10000 reviews, around half of them have no "useful" upvotes, and around 70% of them have no "funny" or "cool" upvotes. Around 30% documents get one "useful" upvote, while the number for "funny"

or “cool” is below 20%. The trend to high upvotes decreases rapidly, showing that only very few reviews are considered very high quality by human annotators. Since a review is more likely to be considered useful than funny or cool according to the graph, the continuing LDA model will focus on number of useful upvotes. The distribution of upvoted documents obviously has a long tail, and it is not surprising that in today’s completive jungle of Internet (even for review recognition!), it’s the very few good reviews that can “survive”. Figure 2 is a partially magnified graph to show the distribution of middle x-axis on a more clear scale.

**Figure 2. Number of Documents Sorted by Upvote Counts (Partially Magnified)**



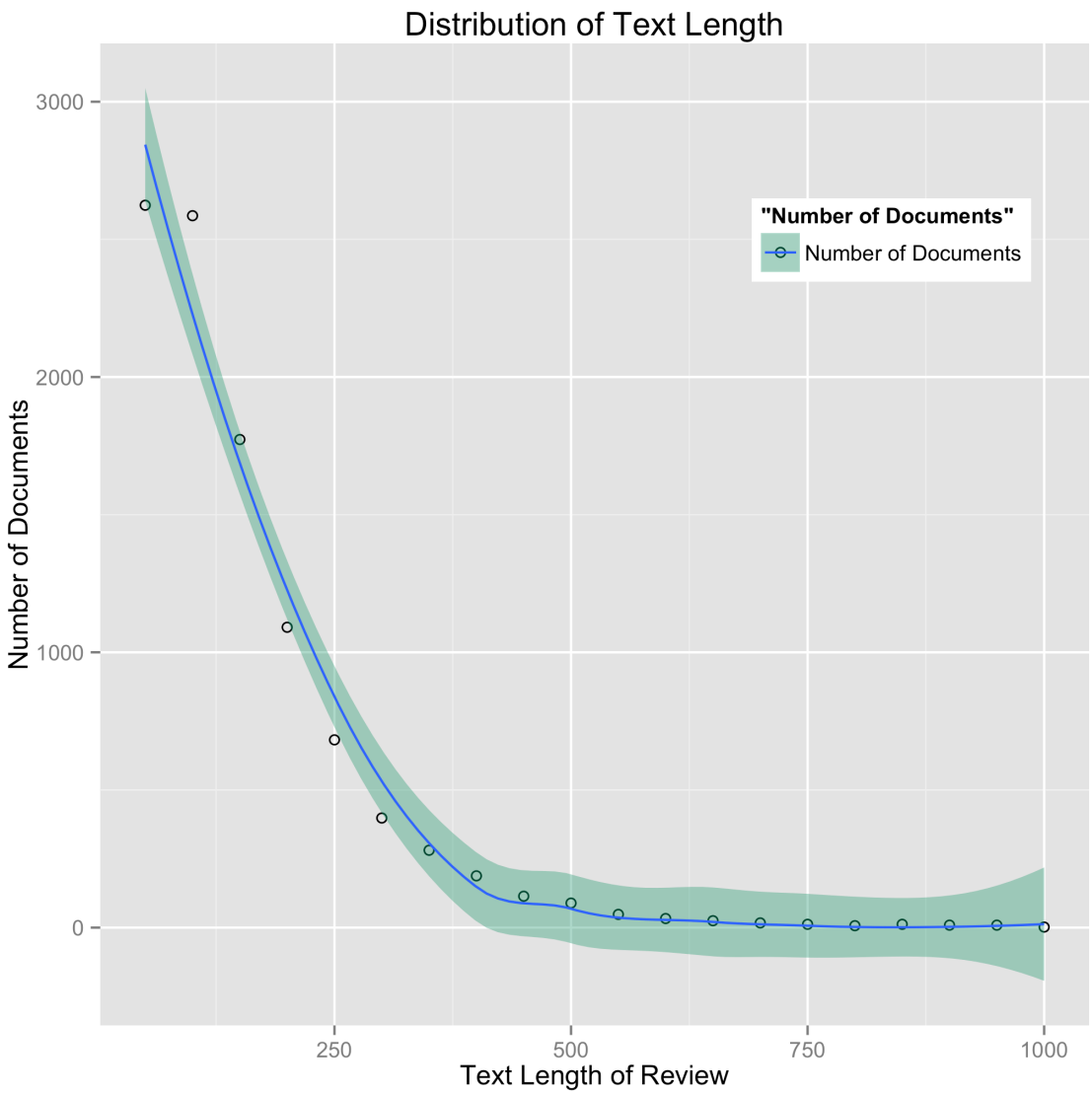

Below (Table 4) is that overview statistics of the sample data. The median text length is 95, showing that more than half of the reviews are under 100 words. The longest review is about a 1,000 word long. It is quite intuitive that shorter reviews can be considered less useful compared to long, structured ones for the nature of language. The

absolute majority of Yelp users wouldn't bother to write a medium to long review (Figure 3), which appears to be in consistence with the upvote distribution. Shorter reviews can be directly ignored unless very witty or special. This assumption will also be tested in the following part.

**Table 4. Overview Statistics of the Sample Data**

| Variables | Upvotes - funny | Upvotes – useful | Upvotes - cool | Stars | Date (days) | Text length |
|---|---|---|---|---|---|---|
| **Mean** | 0.455 | 1.149 | 0.561 | 3.744 | 7/1/12 | 129 |
| **Median** | 0 | 0 | 0 | 4 | 7/3/12 | 95 |
| **Min** | 0 | 0 | 0 | 1 | 1/1/12 | 1 |
| **Max** | 28 | 33 | 32 | 5 | 12/31/12 | 972 |
| **Standard Deviation** | 1.634 | 2.125 | 1.712 | 1.299 | 636.293 | 117.52 |

**Figure 3. Distribution of Text Length**




### 4.3. Do Later Reviews Get Less Upvotes?

The graph (Figure 4 & Table 5) below checks if reviews created later in 2012 will get less upvotes than those created earlier in that year by summarizing the average number of upvotes reviews created in each month. From Figure 4, there is no noticeable tendency that later created reviews will get less "useful" upvotes, and vice versa. "Cool" upvotes

shows an ambiguous trend of growing. The highest point for useful upvotes happens in May and lowest point is January, which rejected the pattern of "accumulation of upvotes". Thus, we can eliminate the time factor in the model for the sample we get from 2012. However, the fluctuation of the upvote number seems to show a pattern of seasonality, which is worth researching in the future.

**Figure 4. Average Number of Upvotes by Month**

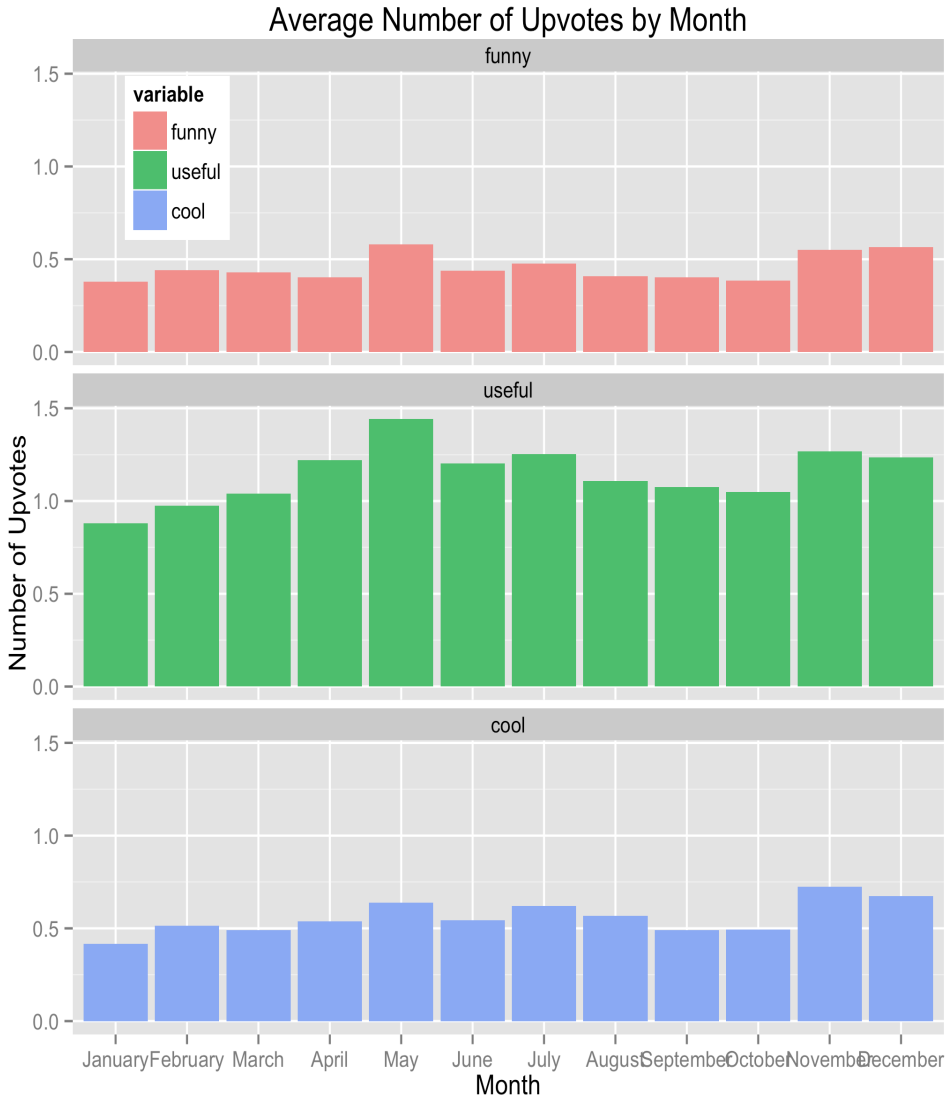


**Table 5. Overview Statistics of Monthly Averages Upvotes in 2012**

| Variables | Upvotes - funny | Upvotes – useful | Upvotes - cool |
|---|---|---|---|
| **Mean** | 0.4552 | 1.1461 | 0.5596 |
| **Median** | 0.4337 | 1.1563 | 0.5416 |
| **Min** | 0.3796 | 0.8802 | 04168 |
| **Max** | 0.5797 | 1.4416 | 0.7326 |
| **Standard Deviation** | 0.0718 | 0.153 | 0.0892 |

### 4.4 Extracting Subtopic of Reviews

With the hypothesis that the more popular subtopics a review best describes, the higher quality it would have and thus, will be decided more "useful" for human annotators. As Yelp provides service for not only restaurants, but also other businesses including clinic, laundry, department store, flower shop etc, words non-related to food should be expected. Before using LDA to subtract subtopics of the Yelp reviews, the paper summarized most frequent words in the reviews to compare with subtopics.

**Table 5. Overview Statistics of Monthly Averages Upvotes in 2012**

| | | | | |
|---|---|---|---|---|
| also | always | around | amazing | back |
| bad | bar | best | better | bit |
| came | can | cant | cheese | chicken |
| come | day | definitely | delicious | didnt |
| dont | eat | even | ever | everything |
| experience | find | first | food | fresh |
| friendly | get | going | good | got |
| great | ive | just | know | like |
| little | love | made | make | meal |
| menu | much | never | new | next |
| nice | night | now | one | order |
| ordered | people | pizza | place | pretty |
| price | really | restaurant | right | room |
| said | sauce | say | see | service |
| since | staff | still | sure | table |
| take | think | time | try | two |
| vegas | wait | want | wasnt | way |
| well | went | will | | |

Obviously lots of frequent words in Table 5 are structural words, prepositions, and conjunction words. The high dimensional LDA model will eliminate these words. The algorithm can also find associated words of a certain word, the example below shows relevant items for “pizza”, “pasta”, “thai”, “laundry”, “seafood”, “wait”, “wifi”, and “return”. There are some non-English words in the relations, such as “docgpizza” or “caash”, due to the algorithm of frequent words.

**Table 6. Associated Words**

| **pizza** | | **pasta** | | **thai** | | **laundry** | |
|---|---|---|---|---|---|---|---|
| crust | 0.42 | boringnot | 0.31 | pad | 0.64 | 116th | 0.41 |
| oven | 0.30 | docg | 0.31 | oldspoiled | 0.41 | 374 | 0.41 |
| slice | 0.29 | docgpizza | 0.31 | contrasts | 0.35 | caash | 0.41 |
| pepperoni | 0.27 | eatif | 0.31 | jerky | 0.35 | footwear | 0.41 |
| pizzas | 0.25 | flavorsit | 0.31 | rediscovering | 0.35 | freezers | 0.41 |
| pizzeria | 0.23 | fonduta | 0.31 | retread | 0.35 | medicaid | 0.41 |
| thin | 0.22 | heavybland | 0.31 | som | 0.35 | trillion | 0.41 |
| **seafood** | | **wait** | | **wifi** | | **return** | |
| ambiancedecor | 0.24 | long | 0.22 | bandwidth | 0.51 | thieves | 0.27 |
| circulate | 0.24 | 1230ish | 0.20 | freeloaders | 0.51 | absolutelybut | 0.24 |
| fishballs | 0.24 | 22person | 0.20 | hogging | 0.51 | apologizedbut | 0.24 |
| pappadeauxs | 0.24 | eggsprobably | 0.20 | login | 0.51 | attachedthe | 0.24 |
| satiate | 0.24 | line | 0.20 | menial | 0.51 | boilshe | 0.24 |
| saucebroth | 0.24 | peaking | 0.20 | outlaw | 0.51 | customersits | 0.24 |
| specifying | 0.24 | seatedand | 0.20 | reconnect | 0.51 | embarrassmenton | 0.2 |

To contains as many topics as the reviews have, for the LDA model we set the K = 100. Below (Table 7) are the 100 topics subtracted from the 10,000 reviews.

**Table 7. Topics Subtracted from 10,000 Reviews from 2012**

| Topic 1 | Topic 2 | Topic 3 | Topic 4 | Topic 5 |
|---|---|---|---|---|
| "quite" | "food" | "lots" | "come" | "like" |
| Topic 6 | Topic 7 | Topic 8 | Topic 9 | Topic 10 |
| "ever" | "around" | "bar" | "loved" | "new" |
| Topic 11 | Topic 12 | Topic 13 | Topic 14 | Topic 15 |
| "big" | "done" | "deal" | "ive" | "feel" |
| Topic 16 | Topic 17 | Topic 18 | Topic 19 | Topic 20 |
| "bit" | "course" | "happy" | "pizza" | "strip" |
| Topic 21 | Topic 22 | Topic 23 | Topic 24 | Topic 25 |
| "order" | "dont" | "person" | "room" | "close" |
| Topic 26 | Topic 27 | Topic 28 | Topic 29 | Topic 30 |
| "service" | "came" | "soup" | "place" | "cool" |
| Topic 31 | Topic 32 | Topic 33 | Topic 34 | Topic 35 |
| "try" | "went" | "next" | "kind" | "actually" |
| Topic 36 | Topic 37 | Topic 38 | Topic 39 | Topic 40 |
| "something" | "anything" | "let" | "people" | "take" |
| Topic 41 | Topic 42 | Topic 43 | Topic 44 | Topic 45 |
| "excellent" | "definitely" | "old" | "meal" | "end" |
| Topic 46 | Topic 47 | Topic 48 | Topic 49 | Topic 50 |
| "chicken" | "store" | "two" | "chips" | "get" |
| Topic 51 | Topic 52 | Topic 53 | Topic 54 | Topic 55 |
| "great" | "super" | "eat" | "thing" | "ask" |
| Topic 56 | Topic 57 | Topic 58 | Topic 59 | Topic 60 |
| "buffet" | "didnt" | "last" | "open" | "said" |
| Topic 61 | Topic 62 | Topic 63 | Topic 64 | Topic 65 |
| "really" | "work" | "best" | "tasty" | "sweet" |
| Topic 66 | Topic 67 | Topic 68 | Topic 69 | Topic 70 |

| | | | | |
|---|---|---|---|---|
| "experience" | "quality" | "cream" | "breakfast" | "need" |
| Topic 71 | Topic 72 | Topic 73 | Topic 74 | Topic 75 |
| "make" | "town" | "back" | "different" | "part" |
| Topic 76 | Topic 77 | Topic 78 | Topic 79 | Topic 80 |
| "many" | "just" | "little" | "burger" | "wait" |
| Topic 81 | Topic 82 | Topic 83 | Topic 84 | Topic 85 |
| "doesnt" | "made" | "amazing" | "perfect" | "got" |
| Topic 86 | Topic 87 | Topic 88 | Topic 89 | Topic 90 |
| "sushi" | "show" | "hot" | "never" | "bread" |
| Topic 91 | Topic 92 | Topic 93 | Topic 94 | Topic 95 |
| "vegas" | "pretty" | "day" | "beer" | "reviews" |
| Topic 96 | Topic 97 | Topic 98 | Topic 99 | Topic 100 |
| "always" | "bbq" | "top" | "love" | "restaurant" |

To calculate the number top topics each review relates to, we set the threshold for gamma in the LDA model to be 0.025. Then, by sorting the most mentioned topics, we get a list of most often-used topics and their frequencies and a word cloud visualization for it (Table 8 and Figure 5).

**Table 8. Most Mentioned Topics**

| Rank | Topic # | Topic | Freq. | Rank | Topic # | Topic | Freq. |
|---|---|---|---|---|---|---|---|
| 1 | Topic 24 | room | 296 | 51 | Topic 76 | many | 77 |
| 2 | Topic 79 | burger | 258 | 52 | Topic 48 | two | 76 |
| 3 | Topic 47 | store | 255 | 53 | Topic 100 | restaurant | 76 |
| 4 | Topic 69 | breakfast | 249 | 54 | Topic 33 | next | 75 |
| 5 | Topic 19 | pizza | 243 | 55 | Topic 80 | wait | 75 |
| 6 | Topic 12 | done | 233 | 56 | Topic 13 | deal | 74 |
| 7 | Topic 64 | tasty | 229 | 57 | Topic 45 | end | 74 |
| 8 | Topic 27 | came | 223 | 58 | Topic 59 | open | 72 |
| 9 | Topic 86 | sushi | 205 | 59 | Topic 67 | quality | 72 |
| 10 | Topic 49 | chips | 194 | 60 | Topic 71 | make | 70 |
| 11 | Topic 96 | always | 193 | 61 | Topic 53 | eat | 69 |
| 12 | Topic 20 | strip | 185 | 62 | Topic 29 | place | 68 |
| 13 | Topic 56 | buffet | 178 | 63 | Topic 84 | perfect | 68 |
| 14 | Topic 51 | great | 175 | 64 | Topic 22 | dont | 67 |
| 15 | Topic 68 | cream | 171 | 65 | Topic 40 | take | 67 |
| 16 | Topic 87 | show | 160 | 66 | Topic 63 | best | 65 |
| 17 | Topic 70 | need | 154 | 67 | Topic 57 | didnt | 64 |
| 18 | Topic 99 | love | 148 | 68 | Topic 62 | work | 64 |
| 19 | Topic 41 | excellent | 143 | 69 | Topic 81 | doesnt | 64 |
| 20 | Topic 90 | bread | 140 | 70 | Topic 82 | made | 64 |
| 21 | Topic 39 | people | 137 | 71 | Topic 16 | bit | 63 |
| 22 | Topic 61 | really | 135 | 72 | Topic 55 | ask | 63 |
| 23 | Topic 28 | soup | 125 | 73 | Topic 88 | hot | 61 |
| 24 | Topic 8 | bar | 122 | 74 | Topic 93 | day | 60 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 25 | Topic 60 | said | 122 | 75 | Topic 54 | thing | 59 |
| 26 | Topic 46 | chicken | 116 | 76 | Topic 58 | last | 59 |
| 27 | Topic 97 | bbq | 115 | 77 | Topic 85 | got | 59 |
| 28 | Topic 18 | happy | 110 | 78 | Topic 4 | come | 58 |
| 29 | Topic 2 | food | 109 | 79 | Topic 32 | went | 58 |
| 30 | Topic 1 | quite | 107 | 80 | Topic 5 | like | 57 |
| 31 | Topic 15 | feel | 107 | 81 | Topic 37 | anything | 57 |
| 32 | Topic 92 | pretty | 103 | 82 | Topic 73 | back | 57 |
| 33 | Topic 44 | meal | 102 | 83 | Topic 75 | part | 57 |
| 34 | Topic 11 | big | 100 | 84 | Topic 78 | little | 57 |
| 35 | Topic 94 | beer | 97 | 85 | Topic 23 | person | 56 |
| 36 | Topic 26 | service | 91 | 86 | Topic 50 | get | 56 |
| 37 | Topic 42 | definitely | 90 | 87 | Topic 52 | super | 55 |
| 38 | Topic 3 | lots | 89 | 88 | Topic 6 | ever | 54 |
| 39 | Topic 83 | amazing | 89 | 89 | Topic 21 | order | 54 |
| 40 | Topic 89 | never | 88 | 90 | Topic 98 | top | 52 |
| 41 | Topic 65 | sweet | 87 | 91 | Topic 36 | something | 51 |
| 42 | Topic 25 | close | 85 | 92 | Topic 9 | loved | 48 |
| 43 | Topic 43 | old | 84 | 93 | Topic 7 | around | 47 |
| 44 | Topic 91 | vegas | 82 | 94 | Topic 74 | different | 47 |
| 45 | Topic 72 | town | 79 | 95 | Topic 14 | ive | 46 |
| 46 | Topic 10 | new | 78 | 96 | Topic 34 | kind | 44 |
| 47 | Topic 31 | try | 78 | 97 | Topic 35 | actually | 44 |
| 48 | Topic 77 | just | 78 | 98 | Topic 66 | experience | 44 |
| 49 | Topic 95 | reviews | 78 | 99 | Topic 38 | let | 43 |
| 50 | Topic 30 | cool | 77 | 100 | Topic 17 | course | 41 |

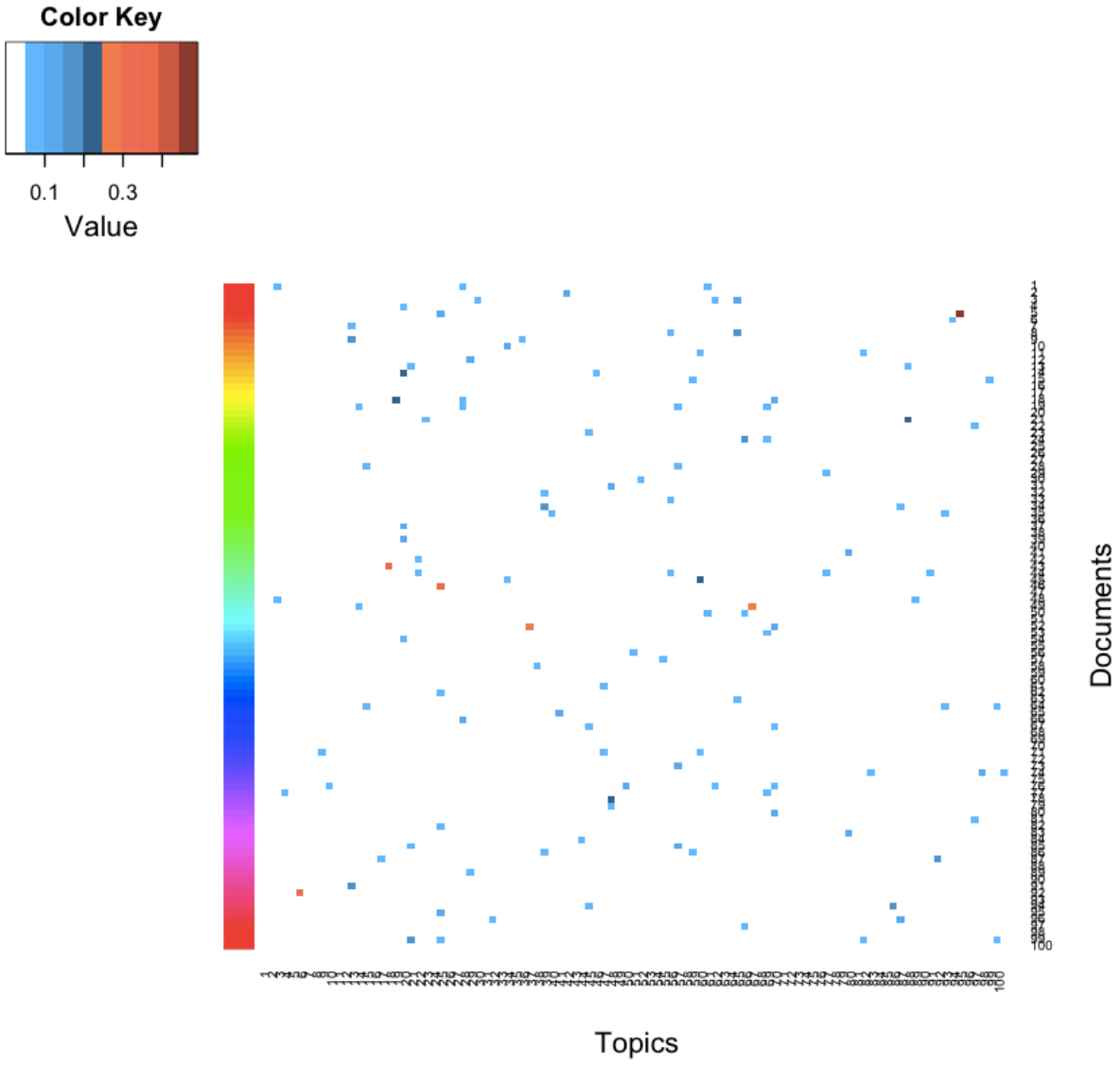
**Figure 5. Word Cloud for Most Mentioned Subtopics**


### 4.5. Subtopic Probabilities

Gamma in the LDA variable represents the probability of a topic representing a document. The heatmap below shows the distribution of the 100 topics among 100

sample documents (out of 10,000 reivews). From the heatmap, we can see most possible topics stay in the probability range of 0 to 0.2, and very few red points show the extremely likely topics of 0.3 to 0.5. There are rows that are plain white, showing that no topics went over the possibility of 0.05 for the according document. In most cases, there are 1 to 5 topics extracted from a document.

**Figure 6. Heatmap for Subtopics Probability Among Documents**



Choosing gamma can be a crucial step for text mining. The tables (Table 9) below show how the distribution of number of documents changes with the change of threshold.

**Table 9. Number of Topic and According Documents (Gamma=0.025)**

| Number of Topics | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Documents | 10 | 123 | 421 | 837 | 1422 | 1646 | 1698 |

| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|
| 1522 | 1119 | 704 | 319 | 133 | 36 | 10 |

**Number of Topic and According Documents (Gamma=0.04)**

| # of Topics | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Documents | 2094 | 4129 | 2709 | 895 | 167 | 6 |

**Number of Topic and According Documents (Gamma=0.02)**

| Number of Topics | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Documents | 35 | 96 | 149 | 212 | 349 | 596 | 810 | 1043 |

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1318 | 1433 | 1283 | 1073 | 716 | 468 | 256 | 103 | 39 | 16 | 4 | 1 |

**Number of Topic and According Documents (Gamma=0.01)**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 15 | 35 | 41 | 84 | 98 | 132 | 125 | 179 | 200 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 250 | 248 | 268 | 352 | 322 | 348 | 360 | 383 | 358 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 360 | 431 | 379 | 400 | 418 | 392 | 430 | 454 | 361 |
| 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 361 | 393 | 334 | 312 | 276 | 224 | 182 | 149 | 100 |
| 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
| 84 | 56 | 37 | 36 | 10 | 11 | 7 | 2 | 3 |

### 4.6 Regression Model

Now it's time to check the correlation between number of topics and useful upvotes with linear regression. The regression takes out variable of word count because we assume it is a confounding factor with the number of topics.

**Table 10. Summary of Linear Regression Model**

| **Residuals:** | | | | |
|---|---|---|---|---|
| **Min** | **1Q** | **Median** | **3Q** | **Max** |
| -1.383 | -1.121 | -0.252 | -0.022 | 31.912 |
| **Coefficients** | | | | |
| | **Estimate** | **Std. Error** | **t value** | **Pr(>\|t\|)** |
| **(Intercept)** | 0.956682 | 0.055916 | 17.109 | < 2e-16 *** |

| | | | | |
|---|---|---|---|---|
| Number of Topics | 0.032792 | 0.008926 | 3.674 | 0.00024 *** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

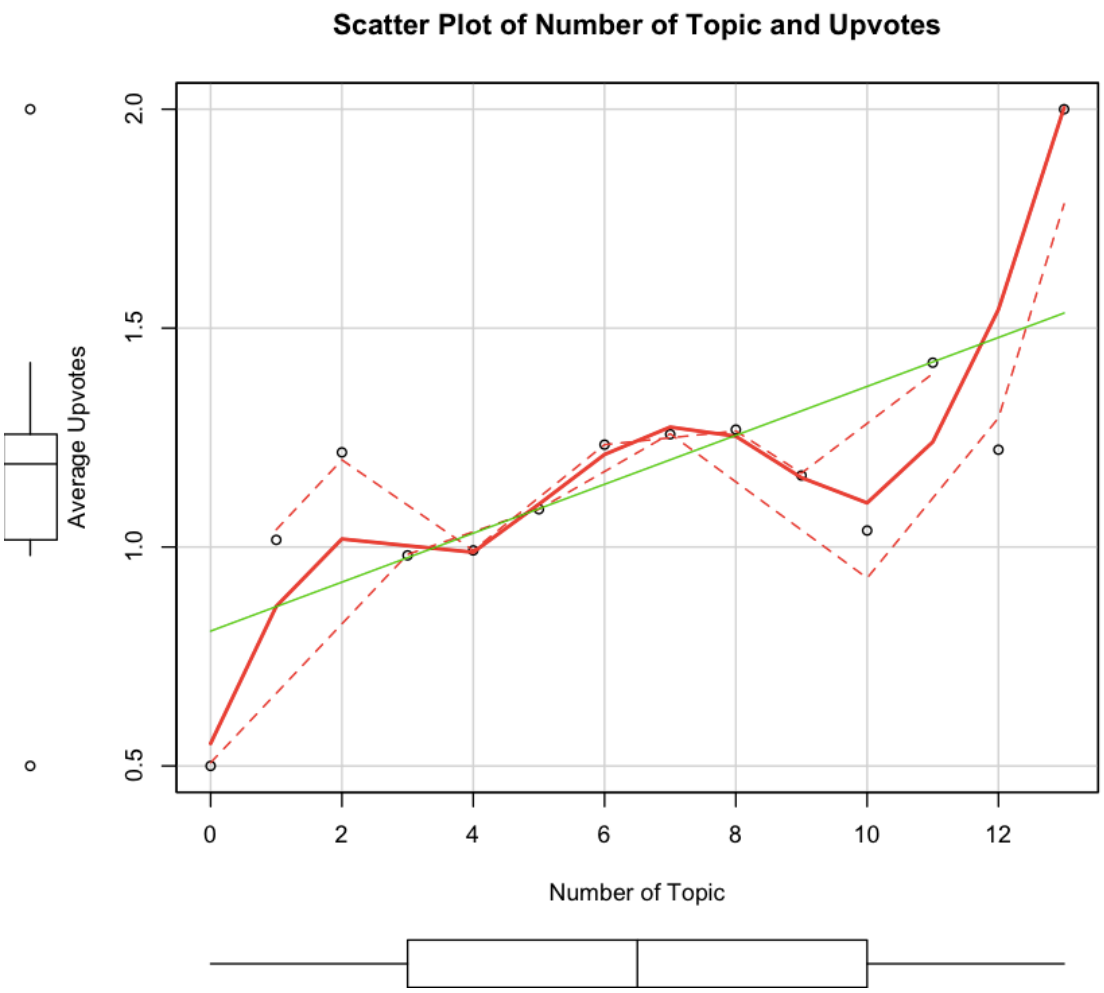
Residual standard error: 1.944 on 9998 degrees of freedom

Multiple R-squared: 0.001348, Adjusted R-squared: 0.001248

F-statistic: 13.5 on 1 & 9998 DF p-value: 0.0002401

---

**Figure 7. Scatter Plot of Number of Topic and Upvotes**



Both regression results and scatter plot shows a strong positive correlation between the number of popular topics a review covers and the useful upvotes it gets.

## 5. Limitations

As broad and deep as topic modeling can get, there are certainly limitations to this paper. There is great space for both methods and dataset to be improved.

First, the default of LDA model used VEM option, while there are other models such as LDA_VEM_fixed, LDA_Gibbs, and CTM model. By running these models with the same dataset and comparing the topic probability distribution of documents, and the trend of perplexity with preset topic number *K* changing. Alpha of LDA and the information entropy are both factors worth looking into as well in future research.

Second, the dataset used in the paper is *very* small for model training, for the limitation of computation ability. LDA is a highly computationally costly model, and a big dataset is the pre-requisite for precise prediction.

Third, the model doesn't take the change of Yelp recommendation system into consideration. With different recommendation mechanism Yelp itself uses, reviews have varied chances of being viewed. The dataset doesn't include internal Yelp recommendation algorithm, which may be a pretty important factor. Other than that, the Yelp academic dataset is a very good one for text mining.

Fourth, the paper doesn't interpret the topics extracted much on a linguistic level. With a good dictionary, those highly interpretable topics can be used to facilitate tasks such as genre discovery and to suggest rating.

## 6. Conclusions

The paper has presented the topic number – upvotes model, which tests the hypothesis that the number of popular topics a review covers and the useful upvotes are positively correlated. The author used LDA_VEM to extract the subtopics of reviews and mined the most popular topics on Yelp. With the confirmed correlation, we can predict the useful upvotes a review will get based on its content. The paper also discovered the seasonality of Yelp review popularity. In addition to that, the paper compared the frequent words and topics clustered by LDA and confirmed the validity of LDA model.

## Reference


Sang, Jitao, and Changsheng Xu. "Faceted subtopic retrieval: Exploiting the topic hierarchy via a multi-modal framework." *Journal of Multimedia* 7.1 (2012): 9-20.

Huang, James, Stephanie Rogers, and Eunkwang Joo. "Improving Restaurants by Extracting Subtopics from Yelp Reviews." (2014).

Berry, Michael W., and Murray Browne. "Email surveillance using non-negative matrix factorization." *Computational & Mathematical Organization Theory* 11.3 (2005): 249-264.

Shinzaki, Dylan, Kate Stuckman, and Robert Yates. "Trust and Helpfulness in Amazon Reviews: Final Report." (2013).

Ramage, Daniel, Christopher D. Manning, and Susan Dumais. "Partially labeled topic models for interpretable text mining." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, 993-1022.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

Chang, J., & Blei, D. M. (2010). Hierarchical relational models for document networks. *The Annals of Applied Statistics*, *4*(1), 124-150.

Hofmann, T. (1999, August). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 50-57). ACM.

Krestel, R., Fankhauser, P., & Nejdl, W. (2009, October). Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on*

*Recommender systems* (pp. 61-68). ACM. Singh, R., & Mukhopadhyay, K. (2011). Survival analysis in clinical trials: Basics and must know areas. *Perspectives in clinical research*, *2*(4), 145. Salton, G., &

McGill, M. J. (1983). Introduction to modern information retrieval. Wang, Y., Sabzmeydani, P., & Mori, G. (2007). Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Human Motion–Understanding, Modeling, Capture and Animation* (pp. 240-254). Springer Berlin Heidelberg.