Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach

Jack Linshi
Yale University
jack.linshi@yale.edu

Abstract

Personalizing Yelp's star ratings relies on the topic modeling processes that allow us to learn the latent subtopics in review text. For example, if a user highly values service but not price, then the star rating can be weighted according to his or her preferences. Whereas traditional topic modeling processes would learn the topics "service" and "price," more beneficial may be to learn the topics "good service" and "bad service," and "good food" and "bad food," assuming that a reviewer would draw from a different set of words when writing about 5-star quality food and 1-star quality food. If reviews are interpreted as justification for star ratings as commonly understood, then equally true is that ratings generate the review content. However, traditional topic modeling lacks methods of incorporating star ratings or semantic analysis in the generative process. In this paper, I propose an approximation of a modified latent Dirichlet allocation (LDA) in which term distributions of topics are conditional on star ratings. I posit that ratings are an approximate function of positively and negatively connoted adjectives. I implement this by adding two diferent "codewords," indicating either the presence of a positive or negative adjective, after each positive and negative adjective in the corpus. In this paper, I first provide exploratory analysis to show how topic term distributions are affected by ratings. Next, I introduce the approximation of this modified LDA, which I call the codeword LDA, and show that when examining documents' topic mixture, this approach produces clearer and more semantically-oriented topics than those of traditional LDA. Finally, I offer examples demonstrating the enhanced topic modeling and predictive powers of this codeword LDA.

I. Introduction

In a Yelp search, a star rating is arguably the first influence on a user's judgment. Located directly beneath business' names, the 5-star meter is a critical determinant of whether the user will click to find out more, or scroll on. In fact, economic research has shown that star ratings are so central to the Yelp experience that an extra half-star allows restaurants to sell out 19% more frequently [1].

Currently, a Yelp's star rating for a particular business is the mean of all star ratings given to that business. However, it is necessary to consider the implications of representing an entire business by a single star rating. What if one user cares about only food, but a particular restaurant's page has a 1-star rating with reviewers complaining about poor service that ruined their delicious meal? The user may likely continue to search for other restaurants, when the 1-star restaurant may have been ideal.

How can we personalize the Yelp experience so that each user receives star ratings that account for his or her preferences? Topic models, such as LDA, allow us to learn the latent subtopics in review texts, which then can be used in various computations to produce a weighted, personalized star rating. My work will focus specifically on the topic model approaches as a means to improve personalization. Though running an LDA model on the corpus of Yelp review texts outputs interpretable topics, this process precludes a fundamental assumption behind Yelp reviews: that reviews justify star ratings, and in turn, that star ratings generate the content of reviews.

In this paper, I propose an approximation of a modified LDA which conditions topics' term distributions not only on the Dirichlet parameter, but also on star ratings. The approximation is based on the assumption that star ratings are an approximate function of adjectives of positive and negative connotations within review text. I show that this approximation produces semantically-oriented topics, such as "good food" and "bad food," which provide better models of latent subtopics and textual semantics than those resulting from traditional LDA. First, I describe the intuition behind using a modified, semantic-driven LDA through exploratory data analysis. Second, I describe the implementation of an approximation of this modified LDA, and compare its output to that of a traditional LDA. Ultimately, if the goal is to create personalized star-ratings, then it is vital that the generated topics closely resemble the topics that a human user would identify.

II. Related Work

In topic modeling, there are several methods for learning abstract topics in a collection of documents. LDA is a common method of unsupervised learning to discover hidden topics. It assumes that there are latent variables that reflect the thematic structure of the documents [3]. Another common topic modeling method is probabilistic latent semantic analysis (PLSI), though it is criticized for not being a proper generative model [4]. A relatively new topic model, the pachinko allocation model (PAM), which models correlation between topics and between words, also appears to be a promising method for

studying Yelp review text as well [11]. Other non-parametric models, such as the Indian buffet process and Chinese restaurant process, which are closely related to PAM, have been used to discover latent hiearchical structures [2].

While there are a number of works that explore the relationship between Yelp review texts and star ratings, most consider text and ratings separately. One related study, for example, uses a traditional LDA to discover hidden topics, and then uses these hidden topics to predict star ratings by averaging the star ratings of all reviews for businesses that contained a particular topic [9]. Another recent study uses unsupervised learning to improve recommendation accuracy and rating prediction accuracy by grouping users together with clustering techniques [7]. One study combines latent rating dimensions and latent review text dimensions, which results in more interpretable topics and more accurate rating predictions [10]. Though my work also revolves around unsupervised learning, my focus is on improving the unsupervised learning process in order to generate topics that better resemble the "true" topic mixture of documents, that is, the topic mixtures that human users would identify.

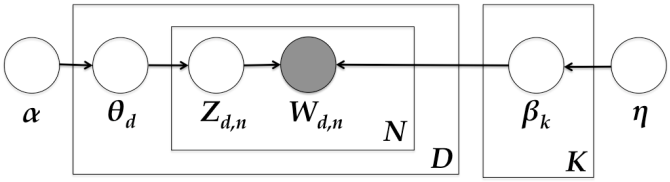
Perhaps most relevant to my work is sentiment analysis, a supervised topic model which has been developed to identify subjective information, such as positive, negative and neutral texts [13]. A more refined sentiment analysis is a feature or aspect-based sentiment analysis, which determines subjective information relating to different aspects [8]. Though sentiment analysis is closely related to my work, which attempts to combine ratings — an indicator of sentiment — with review text, my main contribution is the utilization of text and ratings simultaneously in a single unsupervised learning process.

A relatively new and highly relevant area of study is the intersection between semantic analysis and topic modeling. The first topic model to integrate semantics proposed a word hiearchy for word sampling disambiguation, in which topics can identify the sense of words, contrary to traditional topic models, which do not recognize predefined meanings of words [6]. More recently, another model has been developed that exploits dictionary definitions, which allows the model to better understand semantic relationships within text, and thus produce better textual models [5]. While many of these works create *new* topic models, my work proposes a modified LDA model particularly suitable for the Yelp dataset, then explores a simpler yet effective approximation of this model by examining the theoretical basis for LDA and altering the corpus on which LDA operates.

III. Modeling Ratings and Reviews

Latent Dirichlet Allocation

Traditional LDA can be represented by plate notation (i.e. directed graphical model), which defines the pattern of conditional dependence between the random variables. Latent random variables are depicted by unshaded circles, and observed random variables are depicted by shaded circles. Edges represent dependences between variables, and the rectangular plates represent the number of replications.



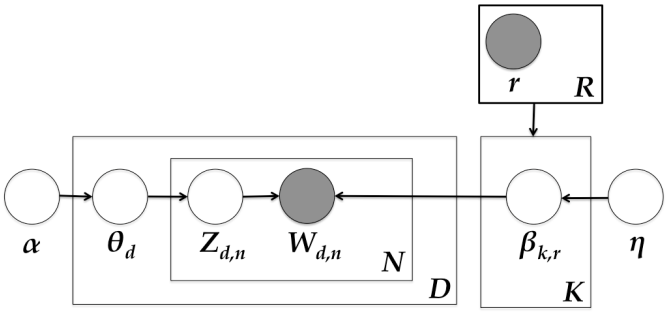
Symbols	
α	Dirichlet parameter
η	Dirichlet parameter
θ_d	topic proportion for d^{th} document
$Z_{d,n}$	topic assignment of n^{th} word in d^{th} document
$W_{d,n}$	n^{th} word in d^{th} document
β_k	distribution of terms in k^{th} topic
N	number of words per document
D	number of documents
K	number of topics

The parameters θ_d and β_k can updated using a variety of methods. Here, the LDA algorithm I utilize relies on collapsed Gibbs sampling, an inference technique that outputs $Z_{d,n}$, θ_d and β_k.

However, traditional LDA does not model the influence that ratings have on β_k. If reviews justify ratings and ratings generate the reviews, and thus the topics, then a more appropriate LDA would model the conditional dependence between a rating r and β_k. In other words, term distributions of a topic would be affected by the value of the star ratings.

Latent Dirichlet Allocation with Ratings

This modified LDA can be represented in plate notation by adding a plate containing the observed variable of ratings.



Symbols	
α	Dirichlet parameter
η	Dirichlet parameter
θ_d	topic proportion for d^{th} document
$Z_{d,n}$	topic assignment of n^{th} word in d^{th} document
$W_{d,n}$	n^{th} word in d^{th} document
$\beta_{k,r}$	distribution of terms in k^{th} topic for the r^{th} rating
r	value of star rating
N	number of words per document
D	number of documents
K	number of topics
R	number of star ratings

Note that the only difference between this plate notation and the previous one (aside from the ratings plate) is that the β_k becomes $\beta_{k,r}$.

Exploratory Analysis

How will the inclusion of the ratings plate alter the topic model output? As a preliminary investigation, LDA is run on a subset of D = 10,000 reviews and K = 10 topics, and then on two subsets of D = 10,000 reviews and K = 10 topics with each subset sampled from exclusively 1-star reviews or 5-star reviews. The review text was pre-processed by stemming words and removing punctuation, extra whitespace, capitalization and stopwords. Tables 1, 2 and 3 show the top 10 words assigned to each topic ranked by probabilities. Each topic has been manually labeled by interpreting the theme represented by the top words.

Tables 2 and 3 demonstrate that when running LDA on only 1-star and only 5-star reviews, the resulting topics contain words that one would use, for example, to describe "good service" or "bad food." Comparing Tables 2 and 3, it is evident that the words assigned to food-related topics (e.g. "Good Food" and "Bad Food") are different: this suggests that topics' term distributions are conditional on rating. Contrastingly, the traditional LDA output in Table 1 shows that adjectives such as "bad" or "good" rarely appear in the top topic words. Thus, if certain documents complained about horrible service experiences, then the topic that Table 1 would have assigned to them (presumably "Service") would likely be less fitting than the topic that Table 2 would have assigned to them (presumably "Bad Service").

Why? LDA aims to identify segregated clusters of co-occuring words, and food-related words (e.g. pizza, chicken, cheese) will co-occur more frequently than food-related words with adjectives (e.g. steak, bad, fries, disgusting). Moreover, because there exists extensive synonyms for these adjectives (e.g. good, great, amazing, wonderful), these words are too diverse to frequently co-occur. More precisely, the joint distribution of the traditional LDA plate notation

$$P(W, Z, \theta, \beta | \alpha, \eta)$$

which is equal to

$$\prod_{k=1}^{k} \underbrace{P(\beta_k|\eta)}_{1} \prod_{d=1}^{D} \underbrace{P(\theta_d|\alpha)}_{2} \prod_{n=1}^{N} \left(\underbrace{P(Z_{d,n}|\theta_d)}_{3} \underbrace{P(W_{d,n}|Z_{d,n}, \beta_k)}_{4} \right)$$

implies that (2) penalizes documents for having too many topics, since small α values result in approximately sparse θ_d matrices. Similarly, (3) penalizes documents with topic mixtures of many topics, since an approximately sparse θ_d matrix would allow a higher probability of assigning a certain topic $Z_{d,n}$ to a word. Both suggest that LDA will try to find topics that represent a diverse number of documents, a diversity that is significantly less when running LDA on subsets of 1-star or 5-star reviews. This discourages creations of topics of exclusively bad or good adjectives (e.g. a "bad adjective" topic consisting of "bad," "nasty," "rude," etc.) because then modeling a review about bad food would require two topics instead of one. Furthermore, (1) indicates that small η values result in approximately sparse β_k matrices, and (4) indicates these approximately sparse β_k matrices increase the probability of certain words $W_{d,n}$. This suggests that sparse topics can be created by identifying clusters of co-occuring words across many documents. Thus, the reason why Table 1 mostly lacks adjectives is because across a subset of all documents of all ratings, these adjectives too rarely co-occur.

It is important to note that while Tables 1, 2 and 3 *all* contain topics describing food, the term distribution for topics is different for 1-star reviews and 5-star reviews: this suggests that when writing about good food or bad food, reviewers draw from different term distributions. However, it makes little intuitive sense to run LDA separately on 1-, 2-, 3-, 4- and 5-star reviews, and then subsequently determine topic mixtures separately by star rating: for example, if a 1-star rating is justified by a review describing good price but terrible quality, it is likely that the 1-star LDA topics, which contain only negative or neutral topics, would classify the review as containing only the "bad quality" topic, and not a "good price" topic. It is also not possible to "combine" the 1-star and 5-star LDA results, as both operate on different corpora and thus different vocabularies. An improved LDA model — one that models term distribution conditional on ratings — would output these topics all at once, such as "good food" and "bad food."

IV. Approximating LDA with Ratings: a Codeword Model

My work revolves around an approximation of the conditional dependence between ratings and topics' term distributions. To do so, I posit that *ratings are an approximate function of adjectives of positive and negative connotations*: a review with mostly positive words would likely be a 4- or 5-star rating, and a review with equal numbers of positive and negative adjectives would likely be a 3-star rating. In fact, a linear regression of star ratings on the number of good words and bad words indicates that on average star ratings can be predicted to within 0.91 stars. It is unlikely that prediction can be significantly reduced to less than 1 star, as sentiment analysis research has shown that human raters agree with a rating 79% of the time: in context, a 21% disagreement rate would correspond to a roughly 1-star average discrepancy on a 5-star scale [12].

Using a list of 203 stemmed positive words and a list of 323 stemmed negative words, I modify the corpus to include a *codeword*, "GOODREVIEW" or "BADREVIEW," after each positive or negative word, respectively. For example, the pre-processed document

> *guy bad car guy awesome car mainten famili servic honest fair priced*

becomes

> *guy bad BADREVIEW car guy awesome*

Breakfast	Greek Food	Service	Food	Flavor	Drinks	Food	Food	Food	Date
little	check	server	food	walk	food	time	time	food	turkey
breakfast	greek	menu	taste	pizza	drink	love	eat	love	theater
eat	pita	nice	service	seat	chicken	sweet	hot	service	film
egg	salad	special	custom	pie	look	friend	soup	delicious	clean
check	try	sandwich	beef	cheese	bar	look	found	time	breast
restaurant	lunch	name	reason	select	employee	serve	store	eat	movie
toast	platter	quick	day	try	taiwanese	try	experience	bar	beer
cookie	souvlaki	look	buy	guess	home	sushi	ramen	star	meat
service	menu	stop	price	return	happy	understand	grill	cheese	told
wait	hummus	feel	noodle	final	hour	tea	lunch	sunday	dinner

Table 1: Top 10 words for K = 10 topics

Bad Food	Bad Service	Bad Wait	Food	Bad Food	Bad Bar	Service	Bad Food	Service	Price
food	store	time	food	pizza	bar	food	food	call	stay
time	time	people	wait	food	drink	time	chicken	car	call
sandwich	custom	flight	time	taste	time	wait	taste	told	people
restaurant	look	line	minute	look	friend	minute	eat	time	hotel
service	rude	hour	table	bad	look	table	rice	day	door
wait	employee	check	drink	eat	pizza	service	dish	custom	day
menu	shop	wait	server	try	people	told	try	service	pay
bad	told	bag	burger	nail	bad	server	sauce	phone	manage
eat	busy	seat	service	worst	wait	manage	fry	manage	time
burger	help	park	eat	taco	night	walk	flavor	tell	money

Table 2: Top 10 words for K = 10 topics for 1-star reviews

Ambiance	Breakfast	Good Service	Good Bar	Good Food	Good Food	Location	Good Food	Good Food	Good Food
bike	breakfast	staff	bar	perfect	salad	try	special	food	delicious
space	pho	high	drink	steak	definitely	location	come	love	love
shape	flavor	recommend	nice	food	super	phoenix	amazing	service	sweet
water	little	owner	enjoy	time	dinner	car	sauce	lunch	fruit
look	egg	incredible	wine	top	dish	lot	restaurant	wait	share
time	gelato	taco	time	amazing	chicken	shape	try	night	slice
try	toast	portion	check	meal	look	restaurant	flavor	day	safe
food	pancake	team	bartend	love	lot	price	garlic	pizza	wings
talk	look	happy	little	yogurt	surprise	live	huge	star	real
filter	tell	perfect	hotel	menu	pleasant	help	bread	excel	friend

Table 3: Top 10 words for K = 10 topics for 5-star reviews

GOODREVIEW car mainten famili servic honest GOODREVIEW fair GOODREVIEW priced

This codeword process enforces co-occurence between adjectives (e.g. delicious, nasty, rude) and nouns (e.g. waiter, coupon, burger) by "standardizing" positive and negative words as "GOODREVIEW" and "BADREVIEW" respectively. In other words, because of the nature of these documents as *reviews*, users are always *evaluating* different aspects of the business: in the same way that the words "bar" and "drink" are likely to co-occur, the words "bar" or "drink" will likely co-occur with critical or complimentary adjectives. The codeword ensures that through the various words users may write to describe good or bad qualities across many documents, they effectively register as one repeated word. In this sense, the codeword is not necessarily *enforcing* co-occurence — it simply makes it evident. Furthermore, by including the codeword in addition to the adjective (as opposed to replacing the adjective with the codeword), this process also enforces co-occurence between codewords and adjectives: it can be shown that if the codewords *replace* the adjectives, the output topics do not appear to be semantically oriented; this may be explained by the fact that adjectives are used much less frequently than nouns, verbs, etc., and the additional codewords "highlight" the semantic elements of the text.

In effect, the LDA algorithm more effectively registers the "presence" of the positive and negative words. By hypothesizing that adjectives (and codewords) are a function of ratings, this roughly approximates the dependence of $\beta_{k,r}$ on r by encouraging the generation of topics such as "good service" and "bad service," which suggests that topics' term distributions depend on ratings. Thus, in the plate notation, r is a function of the number of positive and negative words, an observed quantity, and $R = D$, or the total number of documents, since each review has an associated star rating.

The top topic words from the codeword LDA on a subset of $D = 10{,}000$ reviews and $K = 20$ topics appear in Table 4. For comparison, the results of a traditional model run on the *same* subset D appear in Table 5. In both cases, probabilities have been ranked by a score defined by

$$\beta_{w,k}\left(\log \beta_{w,k} - \frac{\Sigma_{k'} \log \beta_{w,k'}}{K}\right)$$

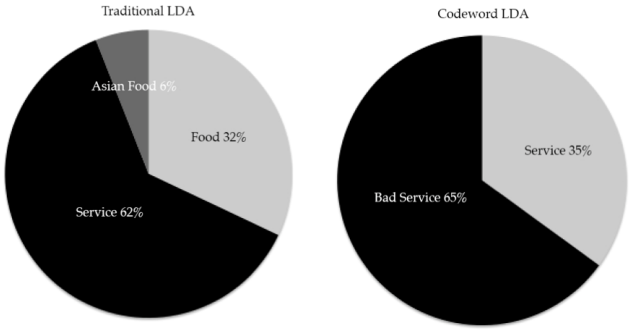
as opposed to simply probabilities, i.e. $\beta_{w,k}$, the probability of the w^{th} word in the k^{th} topic term distribution. This score downweights certain words whose log probabilities are close to the average probabilities for those words across all topic term distributions.

Table 4 indicates that topics from the codeword model have clearer, more distinct topics than those displayed in Table 5, many of which lack obvious, cohesive themes and fail to lend themselves easily to one or two word descriptions. In Table 4, the high probability of "GOODREVIEW" or "BADREVIEW" in a topic suggests the topic is about a good or bad aspect, respectively, and thus a topic in which neither codeword is present can be interpreted as a neutral topic, neither good nor bad. Table 4 also suggests that ratings alter topic term distribution: for example, both "Bad Salon" and "Good Salon" topics are about hair, yet the two topics have very different term distributions. In Table 5, this topic merely appears as "Salon," limiting the specificity of the traditional LDA's topics.

To demonstrate the heightened topic modeling power of the codeword LDA, I consider three randomly selected documents from the pre-processed corpus and compare the topic mixtures from the two LDA models. The review

> *This place is horrible to say the least. There was only one server and she took forever TO get our orders. We were there for a good 20 min before anyone ever came by. Once she did, she took our drink order, took another 20 min to bring our menus, and another 20 min to get our watered down/lukewarm drinks... you get the point. Maybe she was new or something?! We'd see her walk over to the touch screen to place orders, and she'd be there FOREVER trying to figure out how to use it each time. I don't care if it's happy hour all night Thursdays or all week, I will not be back here.*

appears to be entirely about service: namely, bad service. The codeword LDA and traditional LDA have assigned it the following topics:



The codeword LDA appears to have correctly identified the topic mixture entirely about service, particularly identifying it as a document predominantly comprised of the "Bad Service" topic, whereas traditional LDA reports a topic mixture less aligned with the document's human interpretation.

As another example, the review

> *Went here for lunch. The service was great. I liked the chopped salad I shared. I also ordered the Pub burger. It was okay - nothing great. I don't think I'll be rushing back.*

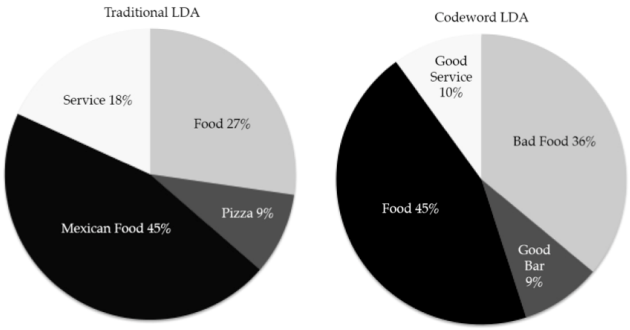
appears to divided between food and service: namely, a bit about good service and mostly about neutral or bad food. This is a particularly interesting example, since this document has one positive topic and one negative (or neutral) topic, and the codeword LDA — which operates on the "bag of words" model, in which order is irrelevant — should theoretically *not* be able to distinguish which topic is good and which topic is negative. The codeword LDA and traditional LDA have assigned it the following topics:

Dessert	Good Service	Bad Salon	Good Price	Good Shopping
ice	wait	BADREVIEW	store	shop
cream	GOODREVIEW	hair	GOODREVIEW	GOODREVIEW
sandwich	wine	bad	shop	love
coffee	food	mean	buy	select
chocolate	server	talk	price	book
cake	minute	worst	grocery	item
flavor	table	rude	love	mall
sub	drink	cut	product	found
cupcake	night	review	coupon	buy
sweet	seat	told	staff	phoenix
Bad Store Service	Good Food	Nightlife	Good Coffee	Bad Food
BADREVIEW	chicken	bar	GOODREVIEW	food
call	salad	pool	coffee	BADREVIEW
custom	GOODREVIEW	night	love	eat
phone	fry	play	park	restaurant
told	sweet	park	super	bagel
store	cheese	stay	shop	egg
card	dish	music	mall	wait
guy	fresh	bathroom	tea	bacon
help	flavor	hotel	spot	menu
line	bread	club	nice	location
Mexican Food	Asian Food	Good Food	Bad Food Service	Bad Travel Service
mexican	sushi	food	BADREVIEW	car
food	food	steak	told	call
taco	chicken	dinner	waiter	airport
salsa	roll	GOODREVIEW	server	BADREVIEW
burrito	rice	meal	call	flight
chip	thai	wine	paid	price
menu	dish	restaurant	card	tire
margarita	pork	dessert	bill	repair
bean	shrimp	salad	check	told
tortilla	sauce	night	water	wash
Food	Good Amenities	Drinks	Good Bar	Good Salon
pizza	GOODREVIEW	beer	bar	nail
burger	stay	food	GOODREVIEW	GOODREVIEW
salad	hotel	game	beer	massage
sandwich	tour	bar	music	spa
fry	course	friend	happy	salon
bread	pool	drink	night	look
cheese	park	seat	fun	pedicure
crust	golf	atmosphere	bike	tip
fries	beautiful	menu	play	hair
chicken	staff	table	live	hot

Table 4: Top 10 words for K = 20 topics from codeword LDA

Dessert	Service	Car Service	Food	Asian Food	Shopping	Park	Food	Breakfast	Salon
cake	told	car	burger	chicken	store	park	burger	salad	store
cupcake	customer	tire	fry	sandwich	coffee	guy	drive	breakfast	hair
wine	manage	service	food	rice	shop	play	line	fresh	nail
red	wait	repair	sandwich	roll	select	bathroom	beer	food	custom
chocolate	minute	replace	menu	flavor	park	hot	bike	menu	cut
drink	call	star	toast	chinese	love	course	park	sushi	salon
enjoy	people	call	french	soup	buy	maybe	ride	chicken	call
park	walk	wash	onion	thai	grocery	run	food	lunch	pay
gift	rude	minute	potato	beef	local	kid	guy	egg	told
popcorn	final	service	cream	spicy	game	people	people	dish	clean
Dessert	Greek Food	Pizza	Food	BBQ	Service	Hotel	Shopping	Bar	Mexican
dessert	chicken	pizza	sandwich	bbq	food	hotel	store	wine	mexican
gelato	salad	crust	lunch	pork	bar	pool	class	bar	food
chef	steak	salad	ice	fry	drink	stay	shop	happy	salsa
menu	food	italian	location	chicken	beer	park	buy	drink	taco
wait	meal	slice	mall	food	waitress	night	doctor	night	burrito
chocolate	bread	pie	menu	beef	table	music	help	hour	chip
cake	pita	cheese	cream	mac	night	play	month	food	flavor
top	hummus	thin	taco	bowl	service	bathroom	care	patio	cheese
table	gyro	tomato	chip	sauce	menu	airport	office	beer	bean
sweet	greek	sauce	fish	meat	waiter	walk	price	dinner	cream

Table 5: Top 10 words for K = 20 topics from traditional LDA

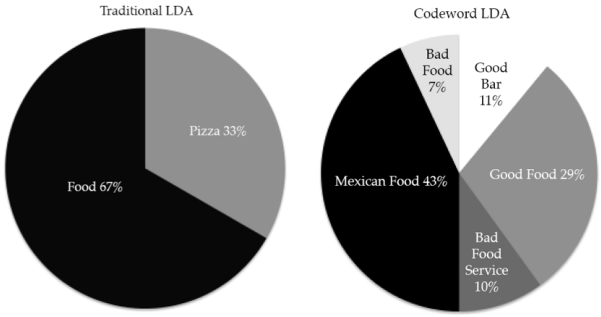


Again, it appears that the codeword LDA model offers a more accurate topical structure: it has, in fact, distinguished that the service was good and the food was bad or neutral. Why? One key observation is that the codeword LDA's topics include "good service," "bad service," "good food" and "bad food," each with distinct term distributions: therefore, the codeword LDA is more able to discern which topic is good and which topic is bad, even if the document is viewed as a bag of words, since the top words of each topic rarely overlap. It also helps that the joint distribution function suggests penalties for documents being assigned too many topics, and by nature of Yelp reviewers frequently writing positively or negatively about *single* aspects, the codeword LDA is able to learn more precisely the word distributions of "good" or "bad" topics by assigning single topics to documents that are, in fact, actually about single topics (e.g. a 1-star review complaining about "bad service," or a 5-star review raving about "good amenities"). This is another reason why codewords do not replace adjectives, as adjectives used to describe service, for example, are different from those used to describe food. Turning to the traditional LDA, it appears that it has classified the document as mostly food, but the wrong type of cuisine. The 9% bar topic that the codeword LDA model identifies is due to the words "pub" and "burger," suggesting the limitation of the codeword LDA's semantic power.

Similarly, as as a final example, the review

> *been here twice and probably won't go back, unless i only have margaritas. the food was great the first time and very sub-par the second, especially for the price of the tacos. the pork ones were especially gross; extremely greasy and fatty and tasteless (i know pork is fatty, but it shouldn't be hard to chew). our service was good the first go round, as we sat at the bar and learned a lot from the bartender about their tequilas. the second time, service was no good. very slow and our waters went empty for entire meal, even after we asked for more.*
> *they do have great guac and chips and their margaritas are fantastic, and they should be for what you pay. but if you are looking for great tacos, skip this place. you'll be underwhelmed.*

appears to be divided between several good and bad topics: good food, bad food, good bar, good service and bad service. The codeword LDA and traditional LDA have assigned it the following topics:



Once again, the codeword LDA appears to have assigned the document a more descriptive set of topics: it, too, has identified the nuances of themes and accurately identified the presence of different sentiments and topics. However, it omits "good service." Still, the codeword LDA offers a far more accurate topic model than that of traditional LDA, which classifies the document in terms of only food.

V. Further Analysis

Since determining the accuracy of topic mixtures of a document requires a human interpretation of the "correct" topic, the codeword LDA does not easily lend itself to efficient evalution of the topic mixtures it assigns. A different method to analyze the effectiveness of the codeword LDA versus the traditional LDA is to regress star ratings on topic mixtures to examine the prediction power of the codeword LDA's topics. Specifically, if the topics of the codeword LDA are semantically oriented, then are the topic mixtures more significant in predicting ratings? Using the $K = 20$ topics from Tables 4 and 5, two $D \times K$ matrices of topic mixtures for each of the $D = 10000$ documents are computed. The vectors of associated star ratings from each corpus of review text are then regressed on each set of 10000 topic mixtures.

Tables 6 and 7 present the regression summaries from the codeword LDA and traditional LDA, respectively. Note that the 20^{th} topic in each table appears as "NA" because of the linear dependence of topic mixtures, i.e., given a document, if the topic proportions of 19 topics are known, then the 20^{th} topic proportion is also known, since the sum of topic proportions for a document must equal 1.

The coefficients in the regression summaries are a clear indication that the topic mixtures from the codeword LDA are far superior in discerning star ratings to those of traditional LDA. In Table 6, estimated coefficients for topics such as "bad travel service" and "good salon" imply that changes in each of these predictor variables, while holding other variables constant, would result in significant decreases and increases in star rating, respectively. In Table 5, very few coefficients have significant p-values, suggesting that topic mixtures generated by a traditional LDA are less powerful in predicting ratings. However, Table 5 still allows us to understand which topics users tend to write about negatively and positively: for example, Table 5 indicates that the "service," "salon," and "car service," which have negative coefficients, are significant predictors at the 5% level for ratings.

Topic	Coefficient	Std. Error	t-value
Intercept	3.4274	0.1768	19.387***
Good Amenities	0.8602	0.2852	3.016**
Bad Travel Service	-1.0531	0.2887	-3.648***
Good Food	1.4319	0.2463	5.813***
Mexican Food	-0.2389	0.27	-0.855
Drinks	0.5475	0.3743	1.463
Nightlife	0.6175	0.3011	2.051*
Good Bar	1.2254	0.3240	3.782***
Good Food	0.8801	0.2379	3.700***
Bad Store Service	-2.8453	0.2644	-10.759***
Food	0.1986	0.3060	0.649
Good Coffee	1.1193	0.2617	4.277***
Asian Food	0.9143	0.2670	3.425***
Bad Food	-1.9723	0.2956	-6.672***
Good Service	0.4272	0.2597	1.645
Good Price	0.6752	0.2672	2.527*
Good Salon	1.6407	0.2859	5.738***
Bad Salon	-0.8514	0.2611	-3.261**
Bad Food Service	-1.3975	0.3034	-4.606***
Dessert	-0.3247	0.3322	-0.977
Good Shopping	NA	NA	NA

*Table 6: Regression summary from codeword LDA (***, **, * represent significance at 0.1%, 1% and 5%, respectively)*

Topic	Coefficient	Std. Error	t-value
Intercept	3.1378	0.4011	7.824***
Park	0.9877	0.7585	1.302
Shopping	1.0145	0.7030	1.443
BBQ	0.5988	0.6477	0.925
Greek Food	1.8795	0.6726	2.795**
Breakfast	1.2760	0.7332	1.740
Bar	0.8789	0.9183	0.957
Dessert	1.3766	0.8566	1.607
Food	0.0705	0.7101	0.099
Hotel	0.2663	0.6943	0.384
Shopping	-0.0110	0.8593	-0.013
Dessert	1.5234	0.7086	2.150*
Pizza	0.2928	0.9017	0.325
Asian Food	0.3626	0.6499	0.558
Food	1.2126	0.7798	1.555
Service	-1.4517	0.67082	-2.164*
Service	-1.5820	0.7601	-2.079*
Salon	-1.48139	0.6122	-2.420*
Food	-0.5095	0.7025	0.725
Car Service	-1.5601	0.7533	-2.072*
Mexican Food	NA	NA	NA

*Table 7: Regression summary from traditional LDA (***, **, * represent significance at 0.1%, 1% and 5%, respectively)*

VI. Conclusion

My work explores topic modeling as it relates to the personalization of Yelp star ratings. I propose a codeword LDA in order to approximate a modified LDA in which term distributions of topics are conditional on ratings. By running two LDAs on subsets of 1-star or 5-star reviews, it becomes evident that it is possible for topics to output topics that describe good, bad or even neutral features — semantic topics that more specifically model document topical structure. My work focuses on using this codeword LDA to insert a codeword after each positive and negative word to encourage LDA to produce these good, bad and neutral topics in *one* unsupervised learning process. In fact, because the number of good words and bad words can be used to predict ratings with relative accuracy, the codeword process effectively fuses ratings *into* the review text. Compared with traditional LDA, this codeword LDA produces topics that are more easily interpretable. With semantically-oriented topics such "good food" and "bad food," the codeword LDA results also show that the codeword process successfully approximates the modified LDA, in that ratings ostensibly alter the term distribution of topics. Finally, the codeword LDA's output can also be used to predict ratings more accurately than that of traditional LDA, not only verifying the effectiveness of codeword LDA as an approximation, but also suggesting further applications of this process. Overall, in the process of Yelp star rating personalization, the codeword LDA produces more accurate and nuanced topics than those of traditional LDA, and presents itself as one step forward in improving the Yelp user experience.

References

[1] M. Anderson and J. Magruder. "Learning from the Crowd." *The Economic Journal*. 2011.

[2] K. Bellar, N. Dalvi, and A. Pal. "Discovering Hierarchical Structure for Sources and Entities." *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. 2013.

[3] D. Blei, A. Ng and M. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research*. 2003.

[4] D. Blei and M. Hoffman. "Online Learning for Latent Dirichlet Allocation." *Neural Information Processing Systems*. 2010.

[5] M. Diab and W. Guo. "Semantic Topic Models: Combining Word Distributional Statistics and Dictionary DeïňĄnitions." Department of Computer Science, Columbia University. 2011.

[6] J. Boyd-Graber, D. Blei, and X. Zhu. "A topic model for word sense disambiguation." *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 2007.

[7] I. Gurevych, N. Jakob, M. Muller, and S. Weber. "Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations." *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*. 2009.

[8] M. Hu and B. Liu. "Mining and Summarizing Customer Reviews." *Proceedings of KDD 2004*. 2004.

[9] J. Huang, E. Joo, and S. Rogers. " Improving Restaurants by Extracting Subtopics from Yelp Reviews.âĂŹâĂŹ Department of Computer Science, University of California âĂŞ Berkeley. 2013.

[10] J. Leskovec and J. McAuley. "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text." Department of Computer Science, Stanford University. 2013.

[11] A. McCallum and L. Wei. "Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations." *Proceedings of the 23rd International Conference on Machine Learning*. 2006.

[12] M. Ogneva. "How Companies Can Use Sentiment Analysis to Improve Their Business." *Mashable*. 2010.

[13] P. Turney. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews." *Proceedings of the Association for Computational Linguistics*. 2002.