ATTRIBUTE SENTIMENT SCORING WITH ONLINE TEXT REVIEWS: ACCOUNTING FOR LANGUAGE STRUCTURE AND MISSING ATTRIBUTES

By

Ishita Chakraborty, Minkyung Kim, and K. Sudhir

May 2019

Revised June 2021

COWLES FOUNDATION DISCUSSION PAPER NO. 2176R2

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS
YALE UNIVERSITY
Box 208281
New Haven, Connecticut 06520-8281

http://cowles.yale.edu/

Attribute Sentiment Scoring with Online Text Reviews: Accounting for Language Structure and Missing Attributes

Ishita Chakraborty, Minkyung Kim, K. Sudhir

Yale School of Management

May 2021

We thank the participants in the marketing seminars at Duke, HKUST, CUHK, ISB, Kellogg, Penn State, NITIE, UBC, UNC Charlotte, University of Texas, UW Madison, UC Berkeley, Notre Dame, UW Marketing Camp, the Yale Quant Marketing Lunch Seminar, the Yale SOM Faculty Seminar, the 2019 ISMS Doctoral Consortium, the 2018 CMU-Temple Conference on Big Data and Machine Learning, the 2019 SICS, the 2019 Management Science Workshop at University of Chile, 2019 IMRC Conference in Houston and the 2019 SAS AI Symposium.

Attribute Sentiment Scoring with Online Text Reviews: Accounting for Language Structure and Missing Attributes

The authors address two significant challenges in using online text reviews to obtain fine-grained attribute level sentiment ratings. First, in contrast to methods that rely on word frequency, they develop a deep learning convolutional-LSTM hybrid model to account for language structure. The convolutional layer accounts for spatial structure (adjacent word groups or phrases) and LSTM accounts for sequential structure of language (sentiment distributed and modified across non-adjacent phrases). Second, they address the problem of missing attributes in text in constructing attribute sentiment scores—as reviewers write only about a subset of attributes and remain silent on others. They develop a model-based imputation strategy using a structural model of heterogeneous rating behavior. Using Yelp restaurant review data, they show superior attribute sentiment scoring accuracy with their model. They find three reviewer segments with different motivations: status seeking, altruism/want voice, and need to vent/praise. Reviewers write to inform and vent/praise, but not based on attribute importance. The heterogeneous model-based imputation performs better than other common imputations; and importantly leads to managerially significant corrections in restaurant attribute ratings. More broadly, our results suggest that social science research should pay more attention to reduce measurement error in variables constructed from text.

Keywords: text mining, natural language processing (NLP), convolutional neural networks (CNN), long-short term memory (LSTM) Networks, deep learning, lexicons, endogeneity, self-selection, online reviews, online ratings, customer satisfaction

INTRODUCTION

Many firms conduct routine tracking surveys on product/service performance on selected attributes chosen by managers that they believe drive overall customer satisfaction (Mittal et al. 1999, Mittal et al. 2001). The summary scores from these surveys are used as dashboard metrics of overall satisfaction and as performance metrics at firms. As surveys are costly, suffer from response biases and get outdated quickly (Culotta and Cutler 2016, Bi et al. 2019), crowd-sourced online review platforms have emerged as an alternative and cheaper source of scalable, real-time feedback for businesses to *listen in* on their markets for performance tracking as well as competitive benchmarking (e.g., Xu 2019, Li et al. 2019). Further, as peer-peer trust and reputation arising from online review platforms gain in importance relative to brand advertising based trust and reputation (Hollenbeck 2018), consumers use review platforms when making choices in many experience goods markets (e.g., Zhu and Zhang 2010, Luca and Vats 2013). Given this, by necessity, many firms now rely on quantitative metrics of attribute level performance from user generated text—they even link performance benchmarking and employee compensation directly to online review performance.[1]

This paper develops a scalable text analysis method to convert open-ended text reviews from online review platforms to produce attribute level summary ratings.[2] This involves solving two novel and challenging sub-problems. First, it requires developing a text mining framework that can convert the rich texture of attribute level sentiment expressed in the text to a fine-grained quantitative rating scale, that not only captures the *valence* of the sentiment, but also the *degree* of positivity or negativity in sentiment. Our model produces significantly higher accuracy classifications for both sentiment valence as well as fine-grained scoring relative to common benchmark methods. As text is increasingly used in social science research (Gentzkow et al. 2019), our results

[1] As an example, see Dubois et al. (2016) for how Accor Hotels use online review based attribute sentiment score cards to manage hotel property employee performance.

[2] Some review platforms such as Zagat, OpenTable and Tripadvisor ask for numerical attribute ratings from reviewers before open-ended text. This may obviate the need to convert text to numerical attribute sentiment scores; but a key disadvantage is that attribute level questions vastly reduce response rates and quality because of the additional time and cognitive costs on the reviewers (Krosnick 1991, Huang et al. 2015). Therefore many large review platforms (e.g., Yelp, Google and Facebook) only obtain an overall rating and free-flowing, open-ended text feedback. Our approach provides attribute level ratings for such platforms.

suggest many commonly used text analysis methods can produce large measurement error when converting text to quantitative sentiment data, leading to biased inference and erroneous substantive insights.

The second problem is that since reviewers self-select which attributes to write about in open-ended text, many attributes will be missing in unprompted reviews. The challenge is to correctly interpret "silence," when a reviewer does not mention an attribute in the review text and impute the correct sentiment to obtain the aggregate attribute level rating. We show that the correct imputations lead to signficant corrections in the average attribute ratings of restaurants. Given that Luca (2016) estimates that a one point change in rating leads to a 5-9% change in restaurant revenues, these corrections are economically significant. Further, behavioral research has long recognized the importance of the right imputation for missing values because people do not ignore missing attributes and often make complex and imperfect inferences from missing data in evaluations. For example, Slovic and MacPhillamy (1974) and Peloza et al. (2015) discuss some common types of wrong inferences—higher weights on common attributes (i.e. attributes for which information is available for all options) or simply proxy missing attribute score with some unrelated attribute score (extra-attribute mis-estimation). Gurney and Loewenstein (2019) provides an excellent review of this topic. While the nature of these inferences may vary, the general takeaway is that missingness usually worsens choice and decision making. Thus, review platforms are very interested in obtaining corrected attribute ratings.[3] Further, as firms begin to increasingly use these text-based attribute ratings for internal feedback, performance measurement and compensation (e.g., Dubois et al. (2016)), obtaining the corrected metrics becomes even more important. We next describe the key challenges involved in tackling these two problems and how we address them.

[3] To assess the value of imputations in our specific context, we show using an mTurk experiment (see Online Appendix Table OA1), that consumer choices are more consistent with their true preferences when attribute level ratings are available. Managers also clearly would prefer their ratings to be valid—to the extent imputations help obtain valid estimates, they would clearly prefer it. We show that our imputations work better than other common imputations on a holdout sample, and that the corrections are large enough to be economically and managerially significant.

Challenges in Attribute Level Sentiment Scoring from Text

Attribute level sentiment scoring from text involves connecting a specific product attribute (e.g., food, service) to an associated satisfaction rating. With fine-grained sentiment scoring, we need to convert text to more than just valence (positive, negative, neutral), but also represent the degree of positivity and negativity (in say a 1-5 point scale). While there has been some work on sentiment scoring of attribute valence (e.g., Archak et al. 2011, Liu et al. 2019), there has been little work on fine-grained attribute scoring—the focus of our paper. We now describe the challenges involved relative to extant work in the literature. We note that the computer science literature in fine-grained sentiment scoring is still evolving and it remains an open problem in natural language processing (Schouten and Frasincar 2015).

Over the last decade, marketing scholars have extensively used text analysis to identify topics, customer needs and mentions of product attributes. Many of these papers have used "bag-of-words" approaches such as the baseline LDA and lexicons—where the identification of attributes and sentiments is based on the frequency of sentiment words. LDA applications include Tirunillai and Tellis (2014), Hollenbeck (2018), Puranam et al. (2017) and Büschken and Allenby (2016).[4] Archak et al. (2011) use a lexicon method to identify attributes and sentiment valence; but do not address fine-grained sentiment scoring.[5] Word frequency based bag-of-words methods are limited in their ability to adequately score attribute sentiments. For example, the sentences "*The food was pretty bad, not good at all.*" and "*The food was pretty good, not bad at all.*" would both have the same word frequencies, but opposite meanings. Or consider the following examples where *sentiment degree* is modified, as in (i) *"horrible," "not horrible," "not that horrible"* and (ii) *"delight," "just missed being a delight"*. When words are just counted as in bag-of-words, making the connections between the key sentiment words "horrible" and "delight" with their degree modifiers

[4] Büschken and Allenby (2020) show that the quality of topics extracted from an LDA model can be improved by incorporating punctuations and conjunctions ("and", "but"), which link sentences. Such a relaxation of the bag-of-words assumption by using auto-correlation in topics across sentences or across sentence parts separated by conjunctions helps better topic identification.

[5] Timoshenko and Hauser (2018) and Liu et al. (2019) use deep learning models that are not based on word frequencies, but their focus is on attribute and valence identification respectively and hence do not need to account for language structure issues that need to be addressed in fine-grained attribute sentiment scoring.

will be difficult, without considering how they are grouped adjacently to form phrases—i.e., spatial structure.

More generally, in NLP, certain types of sentences are considered "hard" for sentiment scoring (Socher et al. 2013). Like the examples above which modified sentiment degree, *negations* often require accounting for adjacent words, i.e., spatial structure to correctly interpret both valence and sentiment degree. Further, other types of sentences such as *long and scattered sentences* and *contrastive conjunctions* require accounting for both the spatial and the sequential structure of language, as the sentiment is distributed and modified across *non-adjacent words* in a sentence. When there are long sentences with sentiments scattered across attributes, being able to make the right association of the sentiment with the attribute becomes a challenge; further sentiments get modified along different parts of a long sentence, and therefore one has to consider these sequences together in inferring sentiment. Contrastive conjunctions–words/phrases like "but," "despite," and "inspite of" can reverse the sentiment of a sentence—on either side of the conjunction. Implied sentiments are challenging because the meaning/sentiment associated with a word lies within a richer context of its usage.

These examples motivate the need to go beyond frequency-based "bag-of-words" approaches and model the structure of language (in terms of phrases and sequences). In our deep learning model, a convolutional layer captures the spatial structure (grouping of adjacent words), and a Long Short Term Memory (LSTM) layer captures the sequential structure (sequence of adjacent and non-adjacent phrases). This allows us to improve our sentiment classification not only in the aggregate on "easy" sentences, but also on the "hard" sentences.

Accounting for Missing Attributes in Attribute Sentiment Scoring

As described earlier, the current literature on topic identification focuses on the frequency of mentions across reviews to identify the most common or novel needs/benefits, attributes desired by consumers/user. The implicit assumption is that topics or attributes that are not mentioned are not important and can be ignored.

We question the premise that *importance* is the primary reason for why an attribute is mentioned or not. There can be other reasons for why a reviewer is silent on an attribute. Some may write only if it can *influence* or be *informative* to readers. For example, if there is high variance among current raters, one's rating can be influential and informative. Or if one's own rating is different from the consensus based on current reviews, one may be motivated to write a different point of view. There could of course be asymmetry in this motivation depending on whether the deviation from consensus is positive or negative. Finally, some raters may choose not to write when the product meets expectations (and rating would have been a three), but only to praise/vent when they are very satisfied or dissatisfied.

We develop a model-based strategy that imputes *missing* sentiment based on observable restaurant characteristics and observable/unobservable reviewer characteristics. We consider and exploit four key features of the available data in this context in developing and identifying the structural model of rating with self-selection: (1) the same restaurant is visited and experienced by multiple reviewers; given that a restaurant provides similar services to all patrons, we assume that all reviewers receive a common latent utility plus idiosyncratic shocks. (2) the same reviewer visits multiple restaurants, this allows us to identify observable reviewer heterogeneity and unobserved heterogeneity in rating styles—i.e. how they map experienced utility to attribute level ratings. (3) all reviewers provide an overall rating, so given multiple observations from a reviewer, we can infer heterogeneous weights of attributes on overall ratings. (4) Finally variables such as informativeness and the need to praise/vent help account for self-selection.

We allow the structural model of rating behavior to account for heterogeneity in rating styles and weights on attributes driving overall ratings. Specifically, we allow for a nonlinear and heterogeneous mapping from experienced utility to attribute ratings using an ordinal logit and a heterogeneous weighting of different attributes to explain the observed overall rating as a regression. The heterogeneity is modeled within a latent class framework. The attribute mention equation (that models attribute writing choice) helps to account for self-selection.We estimate the model using a nested iterative EM algorithm—an inner iteration for the attribute rating imputation, and an outer iteration for the unobserved heterogeneity parameters. The structural model provides insights on reviewer segments and their behavior. The attribute mention equation enables us to assess the above conjectured "drivers of attribute mention (silence)" in reviews. Together the attribute mention equation and the structural model helps with imputation that takes into account heterogeneous and time varying reviewer and restaurant specific factors. We find that there are multiple reviewer segments with different motivations to write reviews—one segment seeks status, another seeks to vent/praise and a third is altruistic or wants to voice their opinion. Interestingly, we find that informativeness and need to vent/praise drives what attributes are mentioned; not attribute importance. We then validate the imputations from our structural model by showing superior performance relative to simpler homogeneous models and other ad-hoc imputation rules on holdout data. Finally, we demonstrate that corrections for attribute mention based on observable and unobservable heterogeneity leads to significant corrections in average attribute ratings for a business.

We note that our problem definition for attribute level ratings abstracts away from issues of (1) selection in *who* chooses to review (e.g., Li and Hitt 2008, Le Mens et al. 2018) and (2) strategic review shading by reviewers and/or fake reviews (e.g., Mayzlin et al. 2014, Luca and Zervas 2016) when aggregating ratings. Reviewer selection/review shading issues are relevant not just for attribute level ratings, but also for overall ratings; as such any approaches to address these issues for overall ratings should also be applicable for attribute level ratings.

Summarizing, our key contributions are as follows: The paper advances the text analytics literature in marketing by addressing the problem of *fine-grained* attribute sentiment scoring; i.e., we not only capture attribute sentiment valence, but also the *degree* of positivity or negativity in sentiment. For this, we highlight the need to move beyond word frequency based approaches (lexicon and LDA) to a deep learning approach that accounts for language structure. Specifically, we account for the spatial and sequential structure of language using a convolutional-LSTM model. Second, we find that attribute mentions in reviews is driven by need to inform and need to praise/vent, but not based on the importance that the reviewer itself places on the attribute. Using a structural model of rating behavior, we develop a model-based imputation for missing attribute ratings. Overall, we note that though the paper is motivated in the empirical context of online reviews, the problems of generating fine-grained attribute sentiment scoring from text and the interpretation/correction of attribute mention has broad application across many settings. In particular, we note that the large improvements in sentiment scoring accuracy (for both valence and fine grained) using our method suggests that social science research using text analysis (Gentzkow et al. 2019) should pay more attention to advanced NLP methods to reduce measurement error in their constructed variables to avoid biased and misleading inference about tested hypotheses.

The rest of the paper is organized as follows. §2 discusses the related literature. §3 describes the problem of attribute sentiment scoring, the challenges and how our model addresses these challenges. §4 describes the structural model of rating behavior, the estimation strategy, and how the model is used for imputing missing attribute scores. §5 describes our data. §6 summarizes the results. §8 concludes.

RELATED LITERATURE

This paper is related to multiple strands of literature in marketing and computer science. We organize our discussion in two parts.

Text Analytics on UGC and Online Reviews

Table 1 positions our paper with respect to the most relevant literature on online reviews and user generated content in marketing. Some of the early research on user-generated (UGC) content in marketing (e.g., Chevalier and Mayzlin 2006, Dhar and Chang 2009, Duan et al. 2008, Ghose and Ipeirotis 2007, Onishi and Manchanda 2012) uses quantitative metrics like review ratings, volume and word count to infer the impact of UGC on business outcomes like sales and stock prices. While these papers established the importance of studying UGC and its specific role in experience goods markets, they did not investigate content in review text.

Another research stream focused on using UGC content in blogs and review forums to extract insights around customer needs and brand positioning (e.g., Lee and Bradlow 2011, Netzer et al.

2012, Tirunillai and Tellis 2014, Büschken and Allenby 2016). Archak et al. (2011) use UGC to measure sentiment valence (not fine-grained sentiment) on specific product attributes using a lexicon approach and its impact on demand.

Table 1: Most Relevant Marketing Literature on Text Analytics

Paper	Analysis Unit	Sentiment Analysis (Y/N)	Sentiment Granularity	Method	Performance Metric	Attribute Mention (Y)
Godes and Mayzlin (2004) & Chevalier and Mayzlin (2006)	Document	NA	NA	No Text Mining	NA	N
Lee and Bradlow (2011)	Document	N	NA	Bag of Words	Overall	N
Archak et al. (2011)	Document	Y	Binary	Semi-supervised	Overall	N
Netzer et al. (2012)	Document	N	NA	Lexical Networks	Overall	N
Tirunillai and Tellis (2014)	Document	Y	Binary	LDA	Overall	N
Timoshenko and Hauser (2018)	Sentence	N	NA	CNN	Overall	N
Büschken and Allenby (2016)	Sentence	N	NA	SC-LDA	Overall	N
Liu et al. (2019)	Document	Y	Binary	CNN, RNN, LSTM	Overall	N
Büschken and Allenby (2020)	Document	N	NA	Autocorrelated LDA	Overall	N
This paper	Sentence	Y	5-level	Convolutional-LSTM	Overall & Hard Sentences	Y

Fine-grained sentiment analysis for individual attributes is one of the more challenging variants of the sentiment analysis problem (Feldman 2013, Wang et al. 2010). Wang et al. (2010), Taboada et al. (2011) are highly interpretable, but rely on carefully hand-crafted features. They are therefore not scalable. They under-perform in detecting sentiments in “hard” sentences. Supervised text classification methods like SVM (Joachims 2002) do not need hand-crafting and are scalable but they need large amounts of labeled training data (tagged by humans) to reach desired levels of accuracy. Hence deep learning models (Kim 2014, Socher et al. 2013, Zhou et al. 2015) combined with meaning-infused word vectors (Pennington et al. 2014, Mikolov et al. 2013) have revolutionized the field of text mining — they do extremely well on text classification tasks, yet require much smaller volume of training data to attain high levels of accuracy. Thus, they overcome the shortcomings of both traditional supervised as well as unsupervised algorithms. A limitation is that they lack interpretability and so it is hard to understand what is driving the performance of deep learning models. Recently, marketing scholars have used deep learning models for text analysis to answer important questions such as need identification (Timoshenko and Hauser 2018) and the impact of reading reviews about particular attributes on purchasing decisions (Liu et al. 2019), but their focus is not on fine-grained sentiment and hence language structure is less important.

We advance the marketing literature on sentiment analysis in two ways: (i) considering fine-grained attribute sentiment scoring and (ii) moving from "bag-of-words" methods like LDA and lexicons to deep learning models that account for structural aspects of language. Hybrid models that combine features of different deep learning architectures can improve performance on hard tasks (Wang et al. 2016); in that spirit, we motivate and construct a hybrid convolutional-LSTM model. Further, to understand the key drivers of model performance, we test our model on various types of hard sentences. In our corpus, nearly half of the sentences are "hard", justifying the need to account for language structure. By reporting performance metrics not just overall, but on types of "hard" sentences, we offer new benchmarks for performance evaluation in future research.

Missing Attributes (Attribute Mention) in Reviews

Our study of attribute mention in text reviews is primarily related to the statistics literature on missing data and imputations. Rubin (1976) laid the seminal framework for analysis of missing data, in which every data point has some likelihood of missing. Rubin classifies missing data problems into three groups: "Missing Completely at Random" (MCAR), "Missing at Random" (MAR), and "Missing Not at Random" (MNAR). MCAR occurs if the probability of missing is the same for all cases, i.e., causes of the missing data are unrelated to the data. This assumption is likely violated in most settings.

Most modern imputation models for missing data are based on the MAR assumption; i.e., the probability of being missing is the same within groups defined by the observed data. Missing data models under MAR assumption are often estimated using Multiple Imputation, or by likelihood methods. Likelihood based approaches either use Bayesian methods or the EM algorithm for estimation. Recently, Athey et al. (2018) proposed matrix completion methods for imputation in big data settings. However to the extent that attribute choice involves self-selection, even with a structural model of rating with observed and unobserved heterogeneity, the problem still belongs to the MNAR class i.e. missigness of a variable is a function of the variable itself even after controlling for other observed and unobserved characteristics e.g., reviewer's decision to rate could be a function of the rating itself in our setting. The most common approach is to then introduce new identifying restrictions by explicitly justifying a model of missingness for the context at hand, and estimate the joint model of missingness with the behavioral model (Little and Rubin 2019, Mohan and Pearl 2018). In this paper, we augment the structural model of heterogeneous reviewer rating behavior with an equation for attribute mention. The structural model also allows for both observable and unobservable heterogeneity in rating styles and linking of attribute ratings to overall ratings. The model allows for a rich heterogeneous nonlinear mapping from experienced utility to five-level rating and weighted mapping of attribute ratings to overall rating behavior. The attribute mention equation accounts for self-selection. We use an EM algorithm to estimate the model, with imputation that allows for both heterogeneity and self-selection to fill in for missing attribute ratings during the EM iterations.

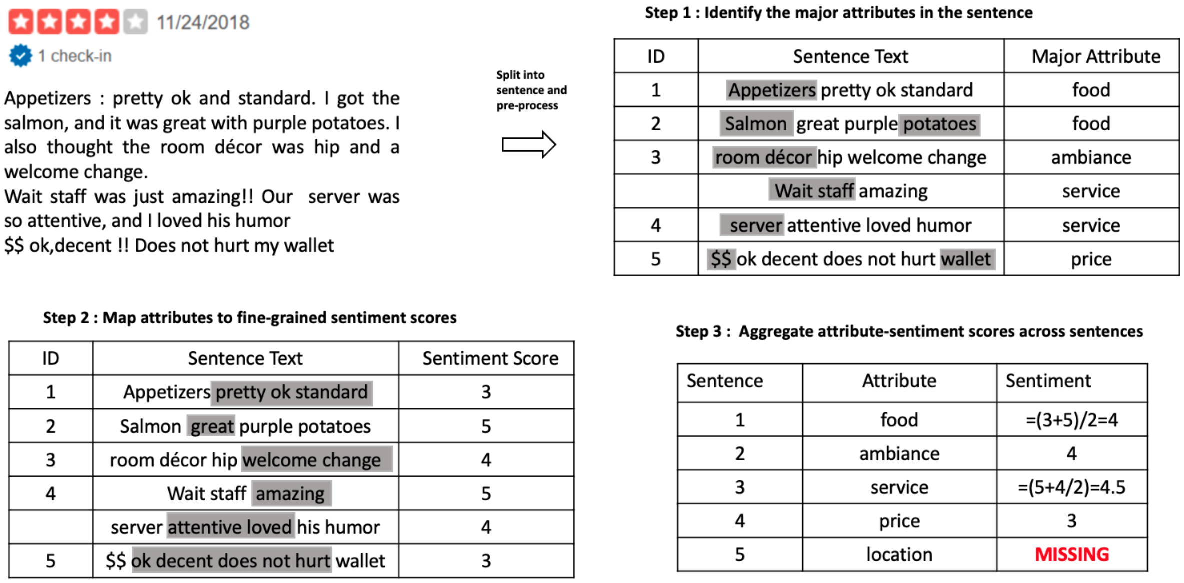
CONVERTING TEXT INTO NUMERIC ATTRIBUTE SENTIMENT SCORES

We first describe the attribute level sentiment analysis problem of converting unstructured text data in reviews into attribute level sentiment scores. We then describe two methods of attribute scoring models with text data: (1) the lexicon model and (2) the deep learning model.[6] Along the way, we also describe various implementation issues and choices that need to be made.

The problem of attribute level sentiment analysis is to take a document d as input (in our empirical example, a Yelp review) and identify the various attributes $k \in K$ that are described in d, where K is the full set of attributes. Having identified the attributes k, the problem requires associating a sentiment score s with every attribute. In solving the attribute level sentiment problem, we make two simplifying assumptions. First, we assume that each sentence is associated with one attribute. Occasionally, sentences may be associated with more than one attribute; in that case, we consider the dominant attribute associated with the sentence. Like Büschken and Allenby (2016), we find that in our empirical setting, multiple attribute sentences account for less than 2% of sentences

[6] For completeness, we also estimate some bag-of-words based supervised machine learning models e.g., Support-Vector-Machine (SVM), Naive Bayes and Logistic Regression as benchmarks, given that they have been used for text classification in the past. We also estimated a supervised topic model S-LDA, but do not report the results as it does not separate attribute and sentiment classes well; a primary requirement for this task.

Figure 1: Illustration of Attribute-Level Sentiment Analysis



Appetizers : pretty ok and standard. I got the salmon, and it was great with purple potatoes. I also thought the room décor was hip and a welcome change.
Wait staff was just amazing!! Our server was so attentive, and I loved his humor
$$ ok,decent !! Does not hurt my wallet

Step 1 : Identify the major attributes in the sentence

ID	Sentence Text	Major Attribute
1	Appetizers pretty ok standard	food
2	Salmon great purple potatoes	food
3	room décor hip welcome change	ambiance
	Wait staff amazing	service
4	server attentive loved humor	service
5	$$ ok decent does not hurt wallet	price

Step 2 : Map attributes to fine-grained sentiment scores

ID	Sentence Text	Sentiment Score
1	Appetizers pretty ok standard	3
2	Salmon great purple potatoes	5
3	room décor hip welcome change	4
4	Wait staff amazing	5
	server attentive loved his humor	4
5	$$ ok decent does not hurt wallet	3

Step 3 : Aggregate attribute-sentiment scores across sentences

Sentence	Attribute	Sentiment
1	food	=(3+5)/2=4
2	ambiance	4
3	service	=(5+4/2)=4.5
4	price	3
5	location	MISSING

in our review data, and thus have very little impact on our results. Second, we assume that the attribute-level sentiment score of a review is the mean of the sentiment scores of all sentences that mention that attribute. We outline the steps involved in obtaining attribute level sentiment ratings from text reviews in Fig 1.

As shown in Fig 1, the first step involves splitting the review text into sentences (all standard sentence separators like full-stops, exclamations and question marks are considered for split). The next steps involve identifying relevant attributes in a sentence and corresponding sentiments. We now describe the process of attribute and sentiment scale selection.

The attribute discovery phase is similar to an exploratory phase before conducting a quantitative survey. For this, we conducted (i) a review of the literature; (ii) an analysis of the most frequent attribute words in the corpus; and (iii) topic modeling using LDA [7]. The literature on restaurant evaluation and industry customer satisfaction surveys identified food quality, employee behavior and wait time (service), basic hygiene, look and feel (ambiance) and value for money as the most common attributes (Ganu et al. 2009). We then did frequent word categorization of our review corpus by associating the most high frequency nouns, noun phrases and select verbs to restaurant-

[7] The LDA results and most important words associated with each topic can be found in the online appendix

relevant attributes. Beyond the four attributes identified from past literature and industry surveys, we found a fifth attribute "location" that has words pertaining to parking, convenience and safety of the restaurant location. Finally, we conducted topic modeling of our review corpus using LDA. As is common with LDA, these topics combined both restaurant attributes and consumer sentiments, and given the very high frequency of food related comments, the topics were disproportionately around food.[8] Overall, we concluded that the five attributes—food, service, ambiance, value and location captured the most relevant attributes for a restaurant rating platform. We use a 1-5 scale for sentiment granularity (1: extremely negative, 3: neutral and 5: extremely positive) as this is comparable to the 5 point rating scale in many review platforms. Also, human taggers fail in practice to differentiate well between classes when the sentiment granularity is higher than 5 levels (Socher et al. 2013). We next describe the two types of attribute sentiment classifiers we consider.

Attribute Sentiment Classifier: The Lexicon Method

We begin with the lexicon-based method because it is highly interpretable, transparent and very widely used and thus serves as useful benchmark relative to more complicated models. The method consists of lexicon construction followed by attribute sentiment classification of text based on dictionary look-ups; i.e. sentences are classified into an attribute and sentiment class by locating word matches in attribute and sentiment class-specific dictionaries. We explain the method below and discuss its limitations.

1. Lexicon building. Lexicon construction involves creating a dictionary of attribute words with corresponding attribute labels (e.g., *waiter*–"service") and sentiment words with sentiment class labels (e.g., *excellent*–"extremely positive"). We first identify the high-frequency attribute and sentiment words in our corpus to create our *vocabulary*. We construct attribute and sentiment class-specific dictionaries, by asking human taggers on Amazon Mechanical Turk to classify all attribute words into one of the five attributes we identified in Step 1—food, service, value, ambiance and

[8] Büschken and Allenby (2016) note that by initializing the LDA model with seed-words for a wider range of attributes, one could obtain more balanced topics. Since we only needed to identify relevant topics and not gain greater balance, using seed-words did not help with identifying additional attributes that were relevant for a large enough set of restaurants to be used on a platform

location and all sentiment words into one of the five sentiment classes–given we decided to use a 5 point rating scale. Every word is labeled by 3 distinct human taggers and we retain only those words for which at least 2 out of 3 taggers agree on the labeling.[9]

2. Attribute Level Sentiment Scoring. Each review is split into sentences. Using the lexicon, each attribute word in the sentence is classified into one of the pre-specified attributes (or none) and each sentiment word is classified into a 1-5 sentiment rating scale using a "look-up" or search of the pre-created lexicons. Following this, the steps are similar to those listed in Figure 1

Despite its simplicity, interpretability and transparency, the method has several limitations. First, lexicon construction is costly in both time and effort, and scales linearly with number of words. Second and more importantly, the method treats language as simply a bag-of-words or "fixed phrases" and does not account for various aspects of language structure. In practice, lexicon methods therefore work fairly well for sentiment identification in simple sentences, but perform poorly on "hard" sentences (Liu et al. 2010).

Why the Lexicon Method Fares Poorly with "Hard" sentences. We elaborate further on why lexicon methods fail to classify *hard* sentences that we had mentioned in the introduction. This is problematic because "hard" sentences are close to 50% of sentences in our review corpus. We now explain each of these types.

1. Negations and Sentiment Degree. Sentences which have different degrees of negative sentiment can be hard to classify without accounting for variable size n-grams. Lexicon methods typically look at one word at a time and will not be able to obtain sentiment valence or degree; Even if ad hoc approaches may be used to address standard negations with bi-grams or tri-grams by hard-coding negation phrases, examples like "*Pizza is not that good*," "*Pizza is not at all great*," illustrate that such ad hoc approaches are unlikely to be effective overall in capturing degree of sentiment. This motivates the use of the convolutional layer, which handles the spatial structure.

[9] While it is possible to use a previously constructed generic lexicon to label attributes and to assign sentiment scores, a domain and task specific lexicon improves classification/labeling accuracy. Moreover, we could not find any existing lexicon that is well-suited for fine-grained sentiment analysis of restaurant reviews. For e.g., AFINN lexicon (Nielsen 2011) and Stanford Sentiment Treebank (Socher et al. 2013) have words and phrases with 5-levels of sentiment classification, however, they are built on Twitter and rotten.tomatoes.com movie review dataset respectively and have limited overlap of words and attributes with our restaurant domain.

2. Long Sentences and Scattered sentiments. In long sentences consisting of more than 20 words, the degree of sentiment (and even polarity) can change multiple times. As an example,“ *OK, in fact good, to start with but kept getting worse and wait staff were unapologetic but manager saved the night*.” In this sentence, the sentiment flows from being good to bad to extremely bad and then back to positive. Yelp reviews tend to have a significant percentage of long sentences. Without sequence history, the classifier cannot capture sentiment shifts and will classify most of these sentences as *neutral* due to the mix of positive and negative sentiment words. More importantly, immediate sentiment modifiers may be changed by sentiment words that are farther away, so having a “long term memory” of what was said before and whether recent sentiment (short-term memory) should take precedence needs to be considered. The LSTM layer helps with both the sequencing and the immediate and distant sentiment modifiers, while the convolutional layer still helps group words into phrases within the long sentence before being fed into the LSTM layer.

3. Contrastive conjunctions. Sentences which have an *X but Y* structure often get misclassified by sentiment classifiers as the model needs to take into account both the clauses before and after the conjunction and weigh their relative importance to decide the final sentiment. An example sentence includes “*Despite the creativity in the menu, execution was a disappointment*.” The first half here is extremely positive due to the word *creativity*, but the second half moderates it significantly. A good classifier should be able to learn from both parts of the sentence to arrive at the correct classification. While the convolutional layer identifies phrases before and after the conjunction, the LSTM layer helps with interpreting the change of meaning after the conjunction.

4. Implied sentiments (sarcasm and subtle negations). These sentences do not have explicit positive or negative sentiment words but the context implies the underlying sentiment. This makes the task of sentiment identification extremely hard for all classes of models and especially for models relying on a specific set of positive or negative words. An example sentence includes “*The place is a treasure if only you are lucky to be there on the right day*.” This is an example of sarcasm, the reviewer uses a positive word like “treasure” but hints at the extreme variance in the type of experience one can have. There could also be subtle negations, for example, “*The girl manag-*

ing the bar had to be the waitress for everyone." Here the reviewer is complaining about lack of service arising out of shortage of staff without using any explicit negative word. Given the meaning/sentiment associated with the word lies in the richer context of its usage, we will empirically assess how much the spatial and sequential structure helps with accurate classification.

Attribute Sentiment Classifier: A Deep Learning Hybrid Convolutional-LSTM Model

Lexicon methods use a constructive algorithm based on pre-coded attributes and sentiment words in a lexicon to score attribute level sentiment. In contrast, deep learning models are a type of supervised learning model, where the model is trained using a training dataset by minimizing a loss function (e.g., the distance between the model's predictions and the true labels). The trained model is then used to score attribute level sentiment on the full dataset. Like deep learning, regression and support vector machines (SVM) are also variations of supervised learning.

What distinguishes deep learning from regression and support vector machines is that deep learning seeks to model high-level abstractions in data by using multiple processing layers (the multiple layers give the name "deep"), composed of linear and non-linear transformations (Goodfellow et al. 2016). Deep learning algorithms are useful in scenarios where feature (variable) engineering is complex and it is hard to select the most relevant features for a classification or regression task. For instance, in our task of fine-grained sentiment analysis, it is not clear which features (combination of variable length *n*-grams) is most informative in order to classify a sentence into "good food" or "great service". The two key ingredients behind the success of deep learning models for NLP are meaningful word representations as input and the ability to extract contiguous variable size n-grams (spatial structure) with ease while retaining sequential structure in terms of word order and associated meaning.

In this section, we outline the architecture of the model and its intuition and discuss critical modeling/implementation choices.[10] Figure 2 shows the general architecture of a neural network used for text classification. Following pre-processing of text, the first layer is the embedding

[10] The technical description is provided in a self-contained online appendix for the interested reader.

layer, where words are converted to numerical vectors by making use of word embeddings. These embedded numerical vectors are then fed to the succeeding feature generating layers, which are the core of the deep learning model. In contrast to older supervised learning methods like SVM which work with the raw data directly as inputs, these feature generating layers, i.e., the convolutional layer and long short term memory network (LSTM) layer in our model, extract higher level features important for classification. The extracted feature vectors are then passed into a logit classifier (soft-max) that classifies the sentence to the class with highest probability of association.

Embedding Layer and Word Representation. Neural network layers work by performing a series of arithmetic operations on inputs and weights of the edges that connect neurons. Hence, words need to be converted into a numerical vector before being fed into a neural network.[11] These vectors are called *embedding* and most well-known embedding algorithms (e.g., word2vec, GloVe) are based on the distributional hypothesis— words with similar meanings tend to co-occur more frequently (Harris 1954) and hence have vectors that are close in the embedding space. The efficiency of the neural network improves manifold if these initial inputs carry meaningful information about the relationships between words. Hence the choice of embedding is an important one — we experiment with both embeddings trained from scratch on our Yelp review corpus as well as a range of pre-trained word embeddings like Word2Vec (Mikolov et al. 2013) and GloVe (Pennington et al. 2014) that are available for all words in our *vocabulary*.[12] There are pros and cons for both approaches — pre-trained embeddings is a form of transfer learning that eliminates embedding generation time, but self-trained embeddings may result in higher classification accuracy due to a more context-relevant vocabulary.

Feature Generating Layers (Convolutional-LSTM). The macro architecture of the neural network comprises of layers to be included (e.g., *feed-forward* or *convolutional*) and type of inter-

[11] The simplest method to form numeric vectors from words is a one-hot representation which means that if there are V words in the vocabulary; each word is represented as a $V \times 1$ dimensional vector where exactly one of the bits is 1 and rest are zero. Such a representation is not scalable for large vocabularies and also stores no semantic information about words. Another option is to only take into account word frequency and simply convert words into numbers based on some normalized frequency score like *tf-idf*.

[12] These embeddings have been trained on different corpus like Wikipedia dumps, Gigaword news dataset and web data from Common Crawl and have more than 5 billion unique tokens.

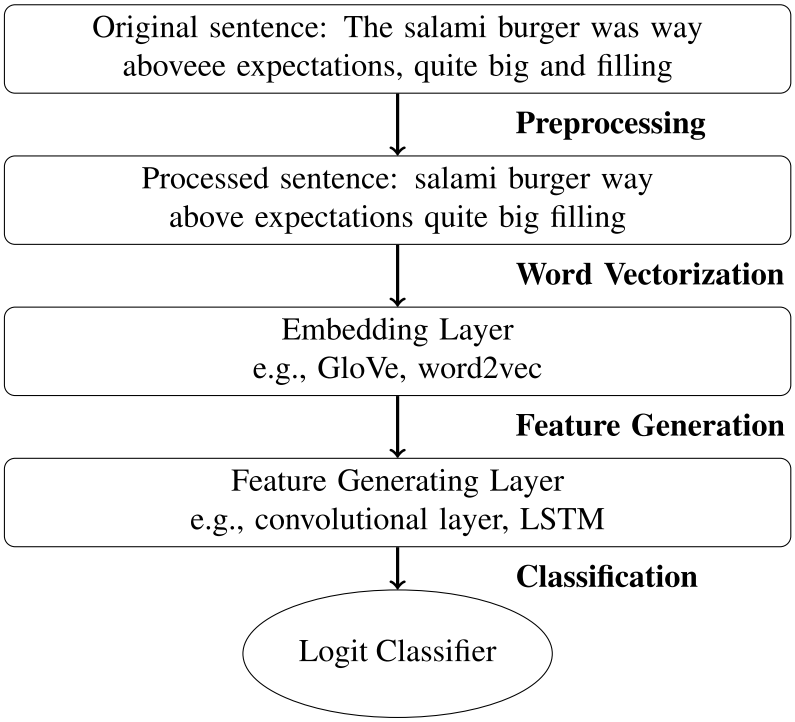
connections between them. As discussed above, the most challenging aspect of our task is dealing with different types of hard negations resulting from variable-size n-grams (e.g., *not good, not that great*) and shifting polarities (*started off well but ended in a sorry surprise*). In many challenging text and image classification problems (Wang et al. 2016), hybrid models that combine the strengths and mitigate the shortcomings of each individual model have been found to improve performance. In that spirit, we build a network consisting of a single convolutional layer with variable-size filters followed by a Long Short Term Memory (LSTM) layer.

Convolutional layers with different filter sizes specialize in extracting variable-length n-grams (phrases) associated with relevant attributes and sentiments and have recently been used successfully in various text analysis applications (Kim 2014, Timoshenko and Hauser 2018). To improve granular sentiment detection where sequence information is critical, we follow the convolutional layer with an LSTM layer that processes the features (phrases) identified from the convolutional layer. LSTM is a variant of the recurrent neural networks (RNN) that specializes in handling longer contextual information (Hochreiter and Schmidhuber 1997). An LSTM employs a cell state (long-term memory) and a combination of gates that are like "regulators" of information to constantly evaluate what parts of the history (in this case n-grams from earlier part of the sentence) need to be forgotten and what needs to be retained to improve the accuracy of the attribute and sentiment classification task.[13] As we motivated in our discussion of "hard" sentences, by taking advantage of the properties of the convolutional layer and LSTM, we expect the hybrid to improve classification accuracy while keeping training time low.

Classifier. The loss function choice depends on the nature of the classification task. Since our tasks involve the classification of text into 5 attribute classes and 5 sentiment classes, it is a multi-class classification problem. We use the standard loss function for multi-class classification called Categorical Cross Entropy. Say s_i represents the convolutional-LSTM model classification for sentence i and t_i represents the ground truth classification, then the cross entropy loss function

[13] For more details on this architecture, see online appendix.

Figure 2: General Architecture of a Deep Learning Network for Text Classification


Original sentence: The salami burger was way aboveee expectations, quite big and filling
Preprocessing
Processed sentence: salami burger way above expectations quite big filling
Word Vectorization
Embedding Layer
e.g., GloVe, word2vec
Feature Generation
Feature Generating Layer
e.g., convolutional layer, LSTM
Classification
Logit Classifier

can be defined in the following manner :

$$Categorical\ Cross\ Entropy\ Loss\ (CCE) = -\sum_{i}^{C} t_i log(s_i)$$

Deep Learning Implementation: Important Choices

Word Embeddings. We tested pre-trained embeddings based on word2vec and GloVe with different numbers of embedding dimensions (e.g., 100, 300) for attributes and sentiment classification. Further, we evaluated whether self-trained embeddings from the specific text corpus can produce superior classification relative to the pre-trained embeddings.

Micro Architecture. The micro architectural decisions in a neural network involve the number of neurons in each of the layers, the size and number of filters for the convolutional layer and dimensions of the max pooling function (that concatenates variable-size feature vectors generated from variable-size convolutional filters). Many of these decisions are empirically driven but some factors that inform these choices are: sentiment classification would rely on presence of long-range n-grams, so we would typically chose a mix of filter sizes for this task ranging from 1-6 grams. In contrast, the attribute classification task often needs only unigrams and bi-grams (*chicken, cola drink, wait time*) and hence simple unigram and bigram filters would be sufficient. Also, since the sequence of *n*-grams matters for sentiment classification, ideally we should not use a max pooling layer after the convolutional layer as the aggregation loses sequential information before being passed to the LSTM layer. However, a pooling layer is needed to merge variable-size feature maps generated from the convolutional filters. We balance this tradeoff by max-pooling on the smallest possible pooling dimension so that we can preserve as much of the sequence information as feasible in sending input into the LSTM layer.

Model Training. As is standard for deep learning models, the model parameters are optimized jointly by training the model iteratively on smaller sub-samples of the training data (mini-batches) and then using the estimation error to improve the model (i.e. change the weights and biases in small increments) through a feedback loop. We experimented with mini-batch sizes of 5, 10, 25, 30, 50 and different optimizers. We chose the RMSProp (Bengio and CA 2015) optimizer because it uses an adaptive learning rate.

Performance Measures for Model Comparison

The primary metric on which we compare our models is accuracy or hit rate. This metric is formally defined as:

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{1}$$

where tp, tn, fp, fn stand for true positives, true negatives, false positives and false negatives respectively. Accuracy is the most common metric that is used for evaluating granular text classification problems and is a fairly good metric unless there is a class imbalance issue (i.e. some classes are not well-represented in the training or test dataset).[14] We also evaluate model performance

[14] While we try to maintain class balance in our data sets, equal representation of all classes is difficult as some classes like food, service appear much more often in Yelp reviews than other classes. Likewise, moderately positive sentiments are more common than extremely positive or negative sentiments.

on specific hard sentence types (e.g., long and scattered sentiments, contrastive conjunctions and implied sentiments) that we discussed earlier in motivating why we account for the spatial and sequential structure of language.[15]

ANALYSIS OF STRUCTURED RATINGS ACCOUNTING FOR MISSING ATTRIBUTES

So far, we focused on converting review text into numerical attribute scores on a 1-5 scale; we coded attributes as "missing" when reviewer is silent on an attribute. For every review, we also have an overall rating on the restaurant. With this quantitative data, we develop and estimate a structural model of reviewer-restaurant experience and rating behavior.

The structural model serves to give us (i) insights into reviewer segments (ii) reviewer attribute rating and writing behaviors and (iii) imputations for missing attributes. We then assess the validity of the model-based imputations on a holdout sample. Finally, we illustrate that corrections for attribute rating using the imputations can be substantial and economically/managerially significant.

A Structural Model of Rating Behavior

The structural model consists of three parts consistent with the data generating process: The first is an ordinal logit model of attribute rating that accommodates (i) a nonlinear mapping from experienced quality to attribute ratings and (ii) heterogeneity in reviewer rating styles. The second is a logit model of attribute mention that allows us to test specific hypothesis related to missing attributes. Using the estimates from the ordinal logit model of attribute rating and the logit models of attribute mention, we impute missing attribute ratings using the Bayes rule. The third part is a regression model of overall ratings against attribute ratings to estimate how attribute ratings impact overall ratings. For this regression, we impute attribute ratings from the previous step when they are missing. The model allows for observed and unobserved heterogeneity. We use an iterative multi-step EM algorithm to estimate the model.

[15] In the online appendix, we report confusion metrics, and derivative metrics such as precision, recall and F1 scores. These metrics give greater insight not just into overall error but also the type of error. Further, we also report qualitative factors such as model building effort, scalability and interpretability for the various models.

We begin with the model of attribute rating. Every reviewer i, who writes a review has an experience with the restaurant. Let A^*_{jk} be the experienced latent quality at restaurant j on the attribute k. The experienced latent utility is a function of observable restaurant quality related characteristics associated with the attribute X^q_{jk} and an idiosyncratic shock that varies across visits. Here, X^q_{jk} includes variables related to restaurant quality like price range, whether the business is a chain or not, average past star rating and average past attribute-level rating.[16] Specifically, let

$$A^*_{ijk} = \alpha_k X^q_{jk} + \nu_{ijk} \tag{2}$$

where ν_{ijk} follows a Type I extreme value distribution (TIEV).

Each reviewer may belong to a segment $g \in \{1...G\}$. The mapping from underlying latent utility A^*_{jk} to the 5 point rating scale A_{jk} can be nonlinear and heterogeneous across reviewers in terms of both observables and unobservables. Specifically, we formulate the nonlinear mapping from latent experienced utility A^*_{ijk} to an ordinal rating A_{ijk} (1-5 scale) as an ordinal logit model given that we assume ν_{ijk} to be TIEV with segment specific g thresholds for each attribute level s:

$$P(A_{ijk} = s) = P(C^g_{k(s-1)} < A^*_{ijk} + \beta_{ik} X^q_i \leq C^g_{ks}) \tag{3}$$

where $C^g_{k(s-1)}$ and C^g_{ks} are the cutoff thresholds of reviewer segment g for attribute k and score s ($C^g_{k0} = -\infty$, $C^g_{k5} = \infty$). The thresholds C^g_{ks} increase monotonically over s and can be heterogeneous across unobserved segments. This allows us to capture unobserved heterogeneity in reviewer's attribute rating style in mapping latent utility to attribute scores. Further, by allowing the cutoff thresholds C^g_{ks} to be a function of observed reviewer specific factors X^q_i (e.g., elite/non-elite), we can also identify how observable factors impact thresholds across different segments. We denote $P(A_{ijk} = s) = \frac{exp(A^*_{ijk} + \beta_{ik} X^q_i - C^g_{k(s-1)})}{1 + exp(A^*_{ijk} + \beta_{ik} X^q_i - C^g_{k(s-1)})} - \frac{exp(A^*_{ijk} + \beta_{ik} X^q_i - C^g_{ks})}{1 + exp(A^*_{ijk} + \beta_{ik} X^q_i - C_{ks})}$ as p_{ijks} for convenience, where $C^g_{k0} = -\infty$ and $C^g_{k5} = \infty$.

[16] This experienced quality can vary over time t as a function of observable restaurant characteristics that vary over time, but for simplicity of notation, we suppress the t subscript in the exposition.

We next formulate the attribute mention model as a binary logit for each attribute for each segment g.

$$P_{ijk}^{Wg} = P^{g}(W_{ijk} = 1) = \frac{exp(\delta_k^g X_{ijk}^w)}{1 + exp(\delta_k^g X_{ijk}^w)} \tag{4}$$

where X_{ijk}^w includes variables that impact the decision of whether to write about an attribute. Given our hypothesis, these include variables related to the incremental informativeness of a review (e.g., variance of past attributes and deviations from existing average rating), but also the attribute sentiment level so that we can allow for reviewers who write to praise or vent; i.e., write only when they want to report an extreme sentiment. By allowing X_{ijk}^w to depend on the sentiment level s (which is related to the latent utility), we allow for a flexible relationship between the experienced utility/reported sentiment in Equation (3) and the choice of writing in Equation (4).[17]

Denote $P_{ijks}^g = P^g(A_{ijk} = s | W_{ijk} = 0, X_{jk}^q, X_i^q, X_{ijk}^w)$, the conditional probability of attribute sentiment being s, when the attribute rating is missing. We can use the attribute mention equation and the ordinal logit model to compute this conditional probability. Omitting the reference to segment g and the conditioning on X_{jk}^q, X_i^q and X_{ijk}^w for clarity, the standard application of Bayes rule $P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|\neg A)P(\neg A)}$, yields

$$P(A_{ijk} = s | W_{ijk} = 0) = \frac{P(W_{ijk} = 0 | A_{ijk} = s) \times P(A_{ijk} = s)}{P(W_{ijk} = 0 | A_{ijk} = s) \times P(A_{ijk} = s) + P(W_{ijk} = 0 | A_{ijk} \neq s) \times P(A_{ijk} \neq s)} \tag{5}$$

Thus the imputation upward weights the probability that the attribute score is a certain level s if it is more likely to be missing at that level s based on the attribute mention equation and under weights it when it is vice versa. Note that all of these weightings will be conditioned on the X variables in the computations.

Finally, we model the overall rating equation for each review as a weighted sum of the ratings

[17] By directly allowing each level of rating of A_{ijk} to have a flexible impact on writing choice, we sidestep the monotonicity assumption typically made when modeling the correlation in error terms between the selection equation and the outcome equation. This allows us to account for the praise/vent hypothesis, where consumers may be more likely to write about an attribute when they receive very low or very high latent utility on that attribute.

on attributes, allowing for both observable and unobservable reviewer heterogeneity (by same latent class as for attribute ratings). Specifically, we model ratings as a segment specific linear regression model:

$$R_{ij} = \gamma_0^g + \sum_k \gamma_k^g A_{ijk} + \varepsilon_{ij} \tag{6}$$

where ε_{ij} follows a normal distribution with mean 0 and variance σ^2.

Then the rating equation with segment specific imputation for missing attribute rating is:

$$R_{ij} = \gamma_0^g + \sum_k \gamma_k^g \left[W_{ijk} A_{ijk} + (1 - W_{ijk}) \sum_{s=1}^{5} s P_{ijks}^g \right] + \varepsilon_{ij}^g \tag{7}$$

where the standard deviation of ε_{ij}^g is denoted as σ^g.

Model Likelihood

The parameters to be estimated are $\Theta = \{\alpha_k, \beta_{ik}, \Theta^1, ..., \Theta^G\}$, where $\Theta^g = \{C_{ks}^g, \delta_k^g, \gamma_0^g, \gamma_k^g, \sigma^g, q_i^g\}$ is the set of segment-level parameters and G is the number of segments. The likelihood function of individual reviewer i belonging to segment g, given $\{\alpha_k, \beta_{ik}, \Theta^g\}$ has three likelihood components.

(8)

$$\begin{aligned}
L_i^g &\equiv L(A_{ijk}, W_{ijk}, R_{ij} | X_{jk}^q, X_i^q, X_{ijk}^w, g; \alpha_k, \beta_{ik}, \Theta^g) \\
&= L(A_{ijk} | X_{jk}^q, X_i^q, g; \alpha_k, \beta_{ik}, C_{ks}^g) L(W_{ijk} | X_{ijk}^w, g; \delta_k^g) L(R_{ij} | P_{ijks}^g, g; \gamma_0^g, \gamma_k^g, \sigma^g) \\
&= \underbrace{\left[\prod_j \prod_k \prod_s (p_{ijks}^g)^{W_{ijk}} (P_{ijks}^g)^{(1-W_{ijk})} \right]}_{L_i^{Ag}\text{: Attribute rating (eq. (3) and (5))}} \underbrace{\left[\prod_j \prod_k (P_{ijk}^{Wg})^{W_{ijk}} (1 - P_{ijk}^{Wg})^{(1-W_{ijk})} \right]}_{L_i^{Wg}\text{: Attribute mention (eq. (4))}} \underbrace{\left[\prod_j \frac{exp(-\frac{1}{2} \frac{\varepsilon_{ij}}{(\sigma^g)^2})}{\sigma^g \sqrt{2\pi}} \right]}_{L_i^{Rg}\text{: Overall rating (eq. (6))}}
\end{aligned}$$

The first component $L_i^{A_g}$ is composed of the likelihood of attribute rating based on the ordinal logit model in equation (3). It consists of (p_{ijks}^g) which is derived only from non-missing values as well as (P_{ijks}^g) which is obtained by updating (p_{ijks}^g) through the Bayes rule using the attribute mention equation (4). The second component $L_i^{W_g}$ is the likelihood of attribute mentions based on equation (4). Here, X_{ijk}^w also includes attribute ratings (that are imputed when missing using estimated α_k, β_{ik} and C_{ks} from the attribute rating ordinal logit model). The third component $L_i^{R_g}$ is the likelihood of the overall rating behavior represented in equation (6). The P_{ijks}^g from the first component determines which rating to be imputed for missing attribute ratings in the rating equation.

The overall likelihood across all reviewers is given by:

$$\sum_i^N ln\Big[\sum_g^G q_i^g L_i^g\Big]$$

There are two challenges with estimating the likelihood above. The first arises from the unobserved heterogeneity making the joint maximization of the likelihood over q_i and L_i^g challenging. We adapt the approach in Arcidiacono and Miller (2011) to estimate the parameters with unobserved heterogeneity. But the more novel second challenge arises from the missing attribute ratings in reviews. To be specific, we have to impute the sentiment rating for missing attributes, i.e., determine P_{ijks}^g, which are jointly used across $L_i^{A_g}$ and $L_i^{W_g}$ and need to be included in the rating equation (6). We describe how we address these issues iteratively in the estimation algorithm next.

Estimation Algorithm

We outline the estimation algorithm here and present the step-by-step details in the appendix. We begin by describing the iterative procedure to maximize the likelihood of the model without unobserved heterogeneity.

First, we estimate the ordinal logit model in equation (3) with only observations that have the attribute ratings reported. This gives us initial estimates of α_k, β_{ik} and C_{ks}. Second, we estimate the attribute mention equation in equation (4), where we use the estimates α_k, β_{ik} and C_{ks} to impute the attribute ratings for reviews when attribute ratings are missing. Then from equation (5), we apply Bayes rule to revise the probability $P(A_{ijk} = s|W_{ijk} = 0)$. We iterate on the estimation of equation (4) and the Bayesian update, until the attribute rating and attribute mention equation estimates converge. This ensures that the attribute rating equation and attribute mention parameters are jointly estimated such that the attribute imputation accounts for selection effects in the attribute mention equation. Finally, we use the $P(A_{ijk} = s|W_{ijk} = 0)$ as inputs into the overall rating equation (6) to estimate the ratings equation.

We then nest the above procedure within another iterative EM procedure as in Arcidiacono and Jones (2003) to estimate the mixture model with unobserved heterogeneity.

EMPIRICAL APPLICATION

Data

Yelp is a crowd-sourced review platform where reviewers can review a range of local businesses e.g., restaurants, spas & salons, dentists, mechanics and home services to name a few. The website was officially launched in a few U.S west coast cities in August of 2005 and subsequently expanded to other U.S cities and countries over the next few years. As of Q1 2017, Yelp is present in 31 countries, with 177 million reviews and over 5 million unique businesses listed (Yelp Investor Relations Q4 2018). Given our empirical application, we focus on restaurant reviews. Since 2008, Yelp has shared review, reviewer and business information for select U.S and international cities as part of its annual challenge. Unique reviewer and business identification numbers in the data helps create a two-way panel of reviews at reviewer and business level. For each review, we observe overall rating, textual evaluation and date of posting as well as information about business characteristics (e.g., cuisine, price range, address, name) and reviewer characteristics (e.g., experience with Yelp, Elite membership). Table 2 summarizes the various data sets we use for different types of analysis. A discussion on each dataset follows.

Table 2: Description of Datasets

Data	**Size**	**Criteria**	**Purpose**
Yelp Restaurant Corpus	1.2 Mn reviews	All restaurant reviews	Exploratory Analysis
Supervised Learning	2400 sentences	Balance of attribute and sentiment classes	Training/Testing Supervised Models
Stratified Sample	45,652 reviews	Business$\geq$20 reviews Mix of Business and Reviewer Types	Estimating Structural Model
Restaurant Panel	250,000 reviews	Restaurants in Stratified Sample	Deriving past review characteristics

1. Exploratory Analysis. We use the full dataset of 1.2 million restaurant reviews for the exploratory analysis to identify attribute and sentiment classes that we described in the model section. We created a *vocabulary* of 8458 words consisting of both sentiment and attribute words.[18] We then did a Parts of Speech tagging of our word list i.e. we classified our word list into adjectives, adverbs, nouns and verbs so as to separate attribute and sentiment words. Attribute words are mainly nouns whereas sentiment words are adjectives and adverbs with some important exceptions: for instance, some verbs are strong indicators of an attribute. e.g, “greeting”, “seated”, “served” refer to *service* and “spent” refers to *value*.[19] Finally human taggers classified the attribute and sentiment words into attribute and sentiment classes. In our dictionaries, we only retain those words that have been labeled into a particular class by at least 2 out of 3 taggers.[20]

2. Training and Test Data for Supervised Learning . For supervised learning, we constructed another data set at the sentence level. Human taggers classify the sentences into its primary attribute and sentiment level. We make sure that the dataset is balanced in its representation of all attribute and sentiment classes. 75% of this data was used for training and the remainder for model validation and testing.

As discussed in §3, lexicon methods do not account for the challenges of obtaining attribute sentiments for hard sentence types. In a randomly sampled subset of sentences from our corpus, 48% of all sentences and 66% of the negative sentences belong to one of the complex types. Long sentences account for 27% of our data. Given their empirical importance, we created a special test dataset of hard sentence types to assess model performance specifically on such sentence types.[21]

3. Restaurant and Reviewer Stratified Sample. To estimate the linkages between attribute level sentiment and overall ratings, we focus on a stratified sample of reviews. We ensure that we have

[18] We excluded stop-words, meaningless phrases and the long tail of words with occurrence frequency less than 1500 in our corpus.

[19] Some adjectives are good indicators of both attribute and sentiment for e.g. the word “cheap” invariably refers to price attribute in a negative way whereas some descriptive adjectives strongly refer to an attribute for e.g., decorated refers to ambiance.

[20] Our attribute and sentiment dictionaries are available upon request. These are more detailed relative to previous studies (Pak and Paroubek 2010, Berger et al. 2010) that focus on two (i.e. positive and negative) or three levels (i.e. positive, neutral and negative) of sentiments.

[21] See online appendix Table OA3 for the composition of training and test data sets and table OA4 for the split of different sentence types.

multiple reviews by individuals so that we can account for unobserved heterogeneity in reviewer rating styles. We want multiple reviews on restaurants to ensure that there are multiple reviewers who obtained similar latent utilities up to a random shock. We therefore restricted our sample to only individuals that posted at least 5 reviews and restaurants that have at least 20 reviews.[22]

We then used stratified sampling by restaurant and reviewer types to ensure that different restaurant types (high and low end; chain and independent) and reviewer types (elite and non-elite; experienced and naive) are represented in the data. This allows us to study how ratings and missing attributes differ by the types.

The sampling leaves us with 45,652 reviews from 2,704 businesses and 19,583 reviewers. As past restaurant reviews might impact current reviews, we incorporate restaurants' time-varying features (e.g., variance and mean of past reviews) by extracting all past reviews for the restaurants in our stratified sample. The full dataset (including all past reviews for restaurants in our sample) contains 250K reviews. We generate each review's time varying variables, including number of past reviews; mean and variance of past star rating; and mean and variance of past attribute ratings.

Table 3 provides some descriptive statistics for reviews, reviewers and restaurants. First, we show a comparison of the characteristics of the full data and stratified sample of 45,652 reviews in terms of review and reviewer characteristics. The mean and median number of reviews per reviewer in our sample is slightly higher than the population (due to stratification). However, the reviewers in our sample are fairly similar to the population in terms of average star rating, experience and length of reviews. The lower part of Table 3 shows the distribution of different business types in the stratified sample. Our sample has almost an equal mix of chain and independent restaurants but independent restaurants get more reviews with higher ratings on average. Low-end and high-end restaurants do not show much difference in terms of average star rating.

[22] The restriction of 5 or more reviews helps eliminate human or bot-generated fake reviews; fakes are mostly from users with one or only few reviews. Luca and Zervas (2016) document that the probability of a user review filtered as spam by Yelp is negatively correlated with the number of reviews by that user.

Table 3: Reviews, Reviewers and Business (Summary Statistics of Dataset and Sample)

	Full			Sample		
Number of Reviews	1.2 M			45,652		
Number of Reviewers	1.02M			19,583		
Reviews and Reviewers						
	Mean	Median	SD	Mean	Median	SD
Star Rating	3.7	3.8	1.09	3.6	3.76	0.92
Review Length	1,109	599	732	709	498	670
Yelp Experience	24	5	82	25.15	17	23.2
Number of Reviews per Reviewer	58	56	27.5	54.6	51.8	36.2
Business						
	All	By Price Range		By Chain		
		Low-end	High-end	Chain	Non-Chain	
Number of Businesses	2,707	1,611	1,096	1,063	1,644	
Number of Reviews	45,652	21,066	24,586	10,528	35,124	
Star Rating Mean (SD)	3.5 (1.4)	3.4 (1.4)	3.6 (1.4)	2.8 (1.5)	3.7 (1.3)	

ATTRIBUTE SENTIMENT CLASSICATION

We first report the results of converting text data into quantifiable attribute and sentiment scores. We report the performance in three parts: (1) Overall classification accuracy; (2) Classification accuracy on “hard” sentence types; (3) polarity and attribute classification.

Overall Classification Accuracy

The lexicon based method that relies on carefully crafted rules and human-tagged lexicons performs better than most supervised machine learning algorithms and is as good as the convolutional-LSTM in the attribute classification task. This is because this task is relatively unambiguous and the lexicons are constructed specific to the domain of restaurant reviews. However, this method does very poorly in the more complex 5-grained sentiment analysis task. Among supervised algorithms, Support Vector Machines (SVM) do better than most of the other classifiers in both attribute and sentiment classification tasks. This is in line with past literature that has shown that SVMs are the best Machine Learning based text classifiers. The network with only convolutional layer just matches the performance of the SVM. However, the convolutional-LSTM does better than all methods in both attribute and sentiment classification tasks. The accuracy of the convolutional-LSTM in the task of 5-level sentiment classification is 50%—lower than state of art accuracy 56% reported in (Brahma 2018), but on a different dataset for which we do not know the differential mix of "hard" versus "easy" sentences in the corpus. Relatedly, other papers do not report dimensions of classification accuracy such as confusion matrices, so we are unable to benchmark on these other relevant accuracy metrics.[23]

Table 4a: Comparison of Text Mining Methods

Type	Method	Attribute accuracy	Sentiment accuracy	Building Effort	Scalability	Interpretability
Lexicon	Lexicon	68%	31%	High	Low	High
Machine Learning	SVM	60%	40%	Moderate	High	Low
	Naives Bayes	43%	39%			
	Logistic Regression	59%	41%			
Deep Learning	CNN	62%	41%	Moderate	High	Low
	LSTM	62%	40%			
	conv-LSTM (pre-trained)	**68**%	**47**%			
	conv-LSTM (self-trained)	**71**%	**50**%			

The convolutional-LSTM model with self-trained embeddings does slightly better than the one using pre-trained Glove embeddings both in terms of attribute and sentiment accuracy. This could be attributed to the slightly more relevant vocabulary generated when word vectors are trained from scratch on a specific corpus.[24]

Classification Accuracy by Sentence Types

To gain intuition about when our model performs better relative to benchmarks, we assess the classification accuracy by sentence type. For this, we sampled 100 sentences of each type from the test dataset. Table 4b reports the results. Our hybrid convolutional-LSTM performs better than other models for all sentence types, but especially so for hard sentences which require considering the spatial and sequential structure. Interestingly, the overall classification accuracy is particularly improved for the scattered sentiments in long sentences.

[23] The state of the art (SOTA) tracking website for NLP, nlpprogress, reports that SOTA for Yelp data for the 5 level sentiment task at the *review document* level is 72% (achieved in 2019). While this task is different from our 5 level *sentence level* sentiment task, to provide a point of comparison, we note that our model's document level performance accuracy (even though we don't optimize around it) is 70%— and comparable to the SOTA from 2017 (e.g., Johnson and Zhang 2017). Interestingly, we use a much smaller training dataset to achieve the same accuracy. While we make no claims in terms of being state of the art in terms of accuracy, we note that our classification results are in the ball park of "best" models. Our focus however here is on improving performance on "hard" sentences, where extant models do not typically do well.

[24] Examples of vocabulary generated with self and pre-trained embeddings are in Table OA6 in online appendix.

Sentiment Polarity and Attribute Classification

A natural question is whether the improvement in CNN-LSTM is only for fine-grained attribute sentiment scoring, and whether it can be of help for sentiment valence as well (i.e. positive, negative and neutral). Overall, the best-performing Convolutional-LSTM model detects the correct valence for 74% of the sentences in the test data compared to 55% for lexicons and 64% for SVM. Further, these improvements are substantially better for various types of hard sentences. The CNN-LSTM with self-trained embeddings is particularly good at preserving polarity for positive classes (4 and 5 ratings) whereas the CNN-LSTM with Glove 300 embeddings is more well-balanced as it preserves polarity reasonably well for both positive and negative classes. Thus our model is potentially valuable even for applications that only require sentiment valence. Also, with respect to attribute classification, the model has more than 70% accuracy across 4 out of 5 classes except location (which is sometimes confused with ambiance). [25]

Table 4b: Performance by Sentence Types

Fine grained (5 level) Sentiment Analysis

	Simple	Hard (Overall)	Scattered	Implied	Contrastive
Lexicon	46%	17%	17%	18%	16%
SVM	47%	19%	18%	20%	20%
CNN	44%	21%	22%	17%	24%
LSTM	46%	30%	37%	28%	25%
Convolutional-LSTM	52%	34%	41%	31%	28%

Sentiment Valence (Positive, Negative, Neutral)

	Simple	Hard (Overall)	Scattered	Implied	Contrastive
Lexicon	66%	44%	48%	39%	44%
SVM	71%	56%	66%	49%	53%
CNN	66%	46%	45%	40%	54%
LSTM	65%	51%	53%	45%	55%
Convolutional-LSTM	86%	64%	80%	48%	64%

STRUCTURAL MODEL OF RATING BEHAVIOR

Overall, the three segment model fits best based on BIC.[26] To help with the interpretation of the structural model estimates, we first describe the descriptive characteristics of the three segments.

[25] We provide these confusion matrices (Table OA4b and OA4c) in the online appendix for the interested reader

[26] The BIC for the 3-segment model at 201 is lower relative to the BIC of the 3-segment and 4-segment models.

We then report and interpret the estimates of the structural model.

Segment Description

Table 5a presents the descriptive statistics of three segments. The smallest Segment 1 (9% of reviewers) consists of 65% elites, writes most often and contributes double their share in reviews (19%). They write the longest reviews, and include the most number of attributes. They tend to write earlier than others on average. They tend to be harsher than the average rating of the restaurants and have relatively low variance of ratings. Given the high percentage of elites, greater frequency, and more comprehensive and longer reviews, we name them as "*status-seeking regulars*."

In contrast, Segment 3 accounting for 30% of reviewers has no elites, writes least frequently, contributing only 25% of reviews. The reviewers write the shortest reviews and include the fewest number of attributes. They tend to write at later stages after others have provided their reviews. They generally tend to be more generous in their overall ratings. Interestingly, they also have the highest variance in their reviews, though they visit restaurants with high ratings and lower variance. We call them the ‘‘*emotive irregulars*," given their lower frequency, and limited contributions in text reviews. They tend to offer either very positive or relatively negative reviews.

Finally, the largest Segment 2 with 61% of the reviewers has only 26% elites and contributes 56% of reviews. Their behavior lies in between the other two more extreme segments. They write fewer, shorter reviews and include fewer attributes than segment 1, but more than segment 3. Their ratings are very similar to the average of the restaurant ratings. We call these reviewers as the "*altruistic mass*." They form the bulk of the Yelp reviewing community who write reviews diligently with little expectation of rewards, and merely want their voice heard.

Mapping Latent Utility to Attribute Ratings: The Ordinal Logit Model

The estimates of the ordinal logit model that maps latent utility to attribute ratings is presented in two parts. The first part of Table 5b presents the mapping between restaurant observables and

Table 5a: Structural Model Segments: Descriptors

	Status-seeking Regulars Mean (SD)	**Altruistic Mass** Mean (SD)	**Emotive Irregulars** Mean (SD)
Share (By Reviewer)	9%	61%	30%
Share (By Review)	19%	56%	25%
N (Reviews)	8,674	25,565	11,413
Characteristics			
% Elites	65%	26%	0%
Review Length (Chars)	889 (744)	677 (640)	349 (329)
No of Attributes	2.87 (1.1)	2.53 (1)	1.87 (0.83)
No of earlier reviews	22 (34.1)	24.2 (38)	36.7 (47.4)
Experience (Months)	33.6 (25.7)	24.6 (25.1)	16.9 (20)
Reviewer Rating	3.9 (0.4)	3.31 (1)	4.08 (1.2)
Business Rating	3.63 (0.7)	3.47 (1.1)	3.84 (0.7)
Proportion of Missing Attributes			
Food	0.10	0.18	0.22
Service	0.22	0.24	0.32
Ambiance	0.53	0.66	0.73
Value	0.58	0.62	0.89
Location	0.71	0.76	0.90

true latent attribute level experience. As expected, restaurants with higher ratings have overall higher latent utility, chains have lower latent utility, and prices reduce latent utility. The thresholds $C_s(s \in 2,3,4,5)$— the cutoff between score $s-1$ and s of the ordinal logit for non-elites for each of the three latent segments are shown in Figure 3. The corresponding graphs are qualitatively similar for Elites, with thresholds for all attribute and segments higher—consistent with Elites being more demanding. As expected, these thresholds are monotone and increasing in rating scale, but nonlinear. Getting higher score requires higher-quality experience across attributes as expected, but the marginal satisfaction required for each score is different across attributes, scores and reviewer segments. It should be noted that even though the thresholds often appear parallel, its implications for probability of a given rating for a segment is highly nonlinear and therefore heterogeneous. This is because there is much higher density in the middle than at the extremes.

Table 5b: Structural Model Estimates (Ordinal Logit and Rating Equation)

Link between Restaurant characteristics and attribute latent utility

Business Characteristic	food	service	ambiance	value	location
Biz price $$	0.171*	0.107**	0.002 ***	0.036***	0.035***
	(0.093)	(0.051)	(0.002)	(0.013)	(0.016)
Biz price $$$	−0.007	0.306***	0.255***	−0.115***	0.356
	(0.084)	(0.0779)	(0.041)	(0.030)	(0.065)
Biz price $$$$	0.100**	0.269**	0.338***	-0.219***	0.650***
	(0.046)	(0.109)	(0.053)	(0.059)	(0.085)
Biz chain	−0.388***	−0.013	−0.079 **	−0.225 ***	−0.351***
	(0.089)	(0.032)	(0.050)	(0.036)	(0.041)
Biz average stars	0.263***	0.395***	0.249***	0.317***	0.254***
	(0.051)	(0.038)	(0.038)	(0.018)	(0.017)
Previous reviews: average attribute rating	0.338***	0.166***	0.046*	0.053***	0.027***
	(0.063)	(0.036)	(0.031)	(0.016)	(0.014)

Attribute Importance by Segment (Normalized Weights)

Attribute	Segment 1	Segment 2	Segment 3
Food	0.229	0.322	0.01
Service	0.273	0.220	0.00
Ambiance	0.131	0.154	0.01
Value	0.177	0.150	0.39
Location	0.188	0.149	0.59

Note: Standard deviations are in brackets *p<0.1; **p<0.05; ***p<0.01

How Attributes Impact Overall Ratings

The second part of Table 5b shows the weights on the attribute ratings that impact overall rating for the three latent segments.[27] Segment 1, (Status Seeking Regulars), the smallest at 9%, places the most importance on food and service in terms of overall ratings. Segment 2 (Altruistic Mass), the largest at 61% cares the most about food. In contrast, the ratings of Segment 3 (Emotive Irregulars) with 30% of reviewers, are driven mostly by value and location.

Attribute Self-Selection in Reviews

Table 5c presents the estimates of the attribute mention equations. The estimates allow us to test the informativenesss and praise/vent conjectures as drivers of attribute mention in reviews.

Informativeness. The first panel of coefficients in Table 5c show support for the informativeness hypothesis. The higher positive attribute coefficients for food and service and negative coefficients of value and location relative to the normalized ambiance coefficient of zero, sup-

[27] For ease of interpretation, the weights reported are normalized such as the sum of the weights add to 1. Note that the model was estimated without normalization and all coefficients were estimated as positive.

Figure 3: Structural Model Estimates:
Attribute Level Thresholds of Latent Utility for Non-elites by Segment

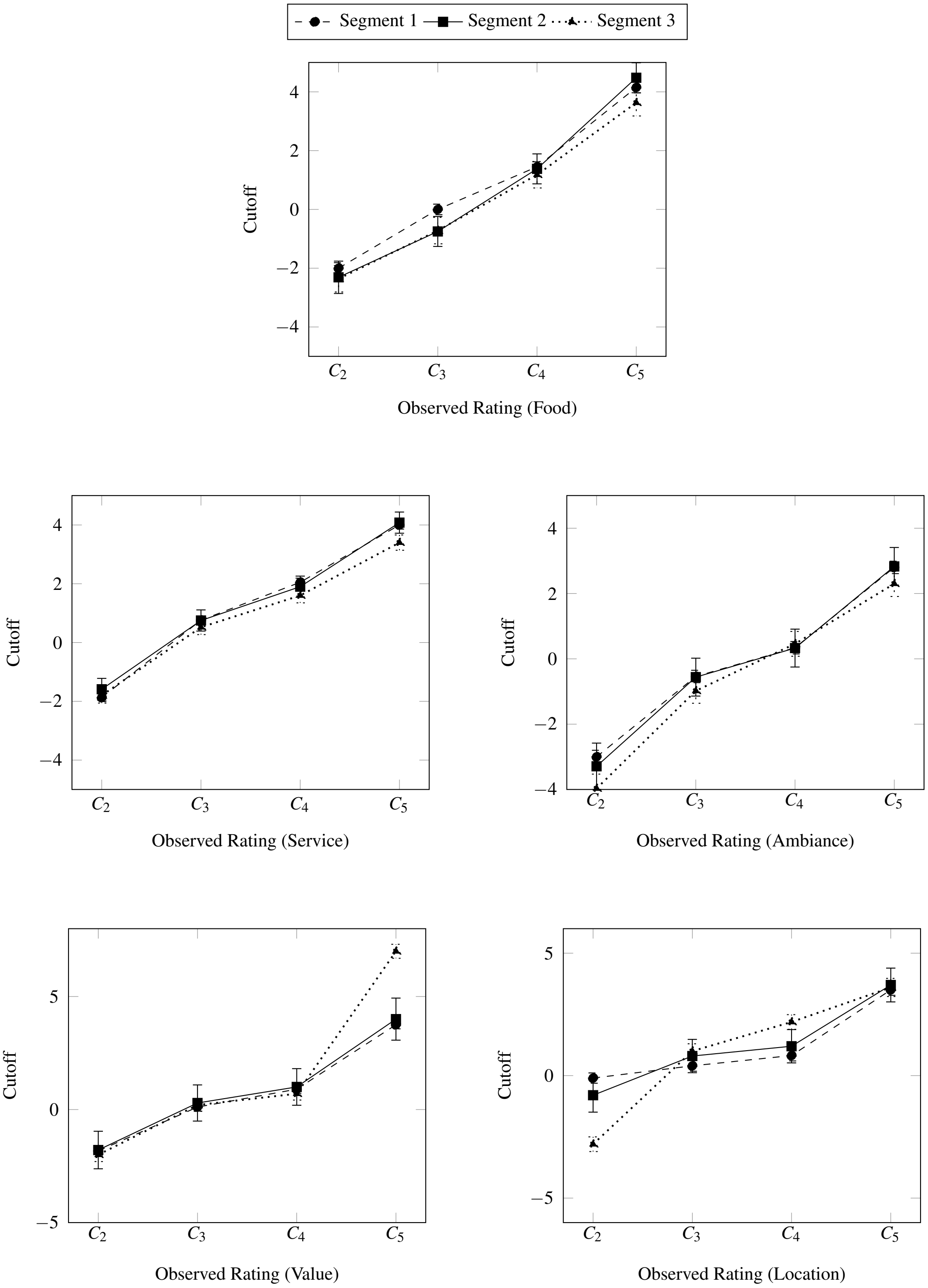


Table 5c: Structural Model Estimates (Drivers of Attribute Mention)

	Dependent variable		
	Attribute Mention (Baseline-Ambiance, Sentiment-Level 3)		
	(Segment 1)	(Segment 2)	(Segment 3)
Informativeness			
Attribute(Food)	0.33*** (0.01)	0.28*** (0.02)	0.58*** (0.02)
Attribute(Service)	0.28*** (0.01)	0.23*** (0.02)	0.53*** (0.02)
Attribute(Price)	−0.20*** (0.01)	-0.65*** (0.02)	-0.003* (0.02)
Attribute(Location)	−0.46*** (0.01)	−0.64*** (0.02)	−0.28*** (0.02)
Chain Dummy	−0.14*** (0.01)	−0.21*** (0.02)	−0.10*** (0.01)
Attribute (Food) ×*Chain*	0.05*** (0.01)	0.04** (0.02)	0.01*** (0.01)
Attribute (Service) ×*Chain*	0.16*** (0.01)	0.29*** (0.02)	0.11*** (0.01)
Attribute (Price) ×*Chain*	−0.03*** (0.01)	−0.01* (0.02)	− 0.06*** (0.01)
Attribute (Location) ×*Chain*	0.16*** (0.01)	0.17*** (0.02)	0.10*** (0.01)
Variance	0.061*** (0.002)	0.059*** (0.004)	0.068*** (0.04)
Negative Deviation	0.077*** (0.005)	0.05*** (0.01)	0.1*** (0.01)
Positive Deviation	−0.03*** (0.002)	−0.03*** (0.004)	−0.03*** (0.004)
Praise/Vent			
Sentiment 1	0.21*** (0.04)	0.21* (0.15)	0.39*** (0.08)
Sentiment 2	0.29*** (0.02)	0.24*** (0.04)	0.50*** (0.03)
Sentiment 4	−0.19*** (0.01)	−0.31*** (0.02)	−0.16*** (0.01)
Sentiment 5	−0.26*** (0.01)	−0.50*** (0.02)	−0.21*** (0.02)
Attribute (Food) × Sentiment 1	−0.33*** (0.04)	−0.28* (0.17)	−0.56*** (0.08)
Attribute (Food) × Sentiment 2	−0.34*** (0.02)	−0.26*** (0.05)	−0.58*** (0.03)
Attribute (Food) × Sentiment 4	0.20*** (0.01)	0.26*** (0.02)	0.07*** (0.02)
Attribute (Food) × Sentiment 5	−0.16*** (0.01)	0.11*** (0.02)	−0.52*** (0.02)
Attribute (Service) × Sentiment 1	−0.32*** (0.04)	−0.26* (0.15)	−0.53*** (0.08)
Attribute (Service) × Sentiment 2	−0.31*** (0.01)	−0.25*** (0.04)	−0.55*** (0.03)
Attribute (Service) × Sentiment 4	0.17*** (0.01)	0.22*** (0.02)	0.11*** (0.02)
Attribute (Service) × Sentiment 5	−0.28*** (0.01)	0.04* (0.02)	−0.45*** (0.02)
Attribute (Price) × Sentiment 1	0.25*** (0.04)	0.68*** (0.17)	0.10 (0.08)
Attribute (Price) × Sentiment 2	0.24*** (0.02)	0.68*** (0.05)	0.09*** (0.03)
Attribute (Price) × Sentiment 4	0.29*** (0.01)	0.75*** (0.02)	0.14*** (0.02)
Attribute (Price) × Sentiment 5	0.02 (0.01)	0.50*** (0.02)	−0.03 (0.02)
Attribute (Location) × Sentiment 1	0.48*** (0.04)	0.66*** (0.15)	0.35*** (0.08)
Attribute (Location) × Sentiment 2	0.39*** (0.02)	0.15*** (0.04)	−0.04 (0.03)
Attribute (Location) × Sentiment 4	0.25*** (0.01)	0.51*** (0.02)	0.18*** (0.02)
Attribute (Location) × Sentiment 5	0.39*** (0.01)	0.53*** (0.02)	0.29*** (0.03)
N	43,370	127,825	57,065
Note: Standard deviations are in brackets			*p<0.1; **p<0.05; ***p<0.01

The baseline for the dummy variables (attribute and sentiment) are attribute ambiance and sentiment 3. We have 45,652 reviews and 5 attributes, hence 228,260 (45,652 × 5) observations (split across segments) to estimate the attribute mention equation.

port our conjecture that reviewers write more often about experience attributes and tend to be more silent on search attributes which can be discovered easily on the site. Further, as expected, variance has a positive coefficient, supporting the hypothesis that attributes are more likely to be mentioned when opinions around that restaurant is not settled. Interestingly, for deviations, negative deviations induce the attribute to be mentioned, but vice versa for positive deviations. This is the case across all segments.

Praise/Vent Need. The second panel of Table 5c based on the coefficients of the interaction terms between attribute and sentiment level help test the praise/vent conjecture for attribute mentions. Those who seek to praise/vent are more likely to report extreme sentiments (1/2 and 5) relative to moderate sentiments (3 and 4). For food and service, there is a higher probability of reporting moderate ratings compared to the more extreme ratings. For price and location on the other hand, there is a tendency to vent especially among Segment 3 who value these attributes a lot.

Finally, we assess the role of attribute importance in attribute mention choice by comparing the probability of missing attribute by segment in the bottom panel of Table 5a with the attribute importance weights of the three segments reported in Table 5b. Food and service (and to a lesser extent ambiance) have the least missing values. Food, service and ambiance also have among the highest impact on overall ratings for Segments 1 and 2. But for segment 3, even though food and service do not drive overall ratings, they still are the most written about attributes. Similarly, even though value and location impact overall rating for Segment 3, these are still the most missing attributes in text reviews. Thus, the relationship between attribute importance and mention is not clear and can vary by reviewer type.

In summary, the information value of reviews play a significant role in the motivation to write about attributes across all segments. We also found the motivation to both praise good performance and vent about bad performance, but this varied across segments and attributes. For staple features like food, service and ambiance, all three segments are more likely to write when satisfied and less likely to write when dissatisfied. Overall, this might explain in general why reviews tend to be skewed to be more positive on rating sites—if this also translates to selection into who writes reviews. However Segment 3 is likely to vent more when dissatisfied about two attributes that drive its ratings—value and location. The lack of a strong link between attribute importance for overall ratings and mentions in reviews suggests that online reviews may not be as complete a source of topic and need identification as previously believed. But we note that this could be because the importance of an attribute for satisfaction conditional on visit may not account for the importance of that attribute in driving visits.

Nevertheless, our results suggest some caution in the use of frequency of mentions as a proxy for benefit or need importance; we suggest that this issue be explored in future research.

Validation of Imputation

We validate our model-based imputation approach in Table 6 by assessing the ability to predict attribute ratings on a holdout sample, relative to some benchmark imputations. For benchmarks, we consider (i) a model with no reviewer heterogeneity in rating styles and rating importance weights, and (ii) ad-hoc fixed imputations. For the fixed imputations, we considered whether missing ratings reflect average satisfaction (3), very low satisfaction (1) or very high satisfaction (5). We do this by comparing the predicted attribute ratings vs. observed rating if an attribute rating is present on hold-out sample (10% of the observations). The overall RMSE across all attributes is lower for our model relative to the benchmarks . Even when the RMSE is compared by attribute, we find that our model does better on all attributes.

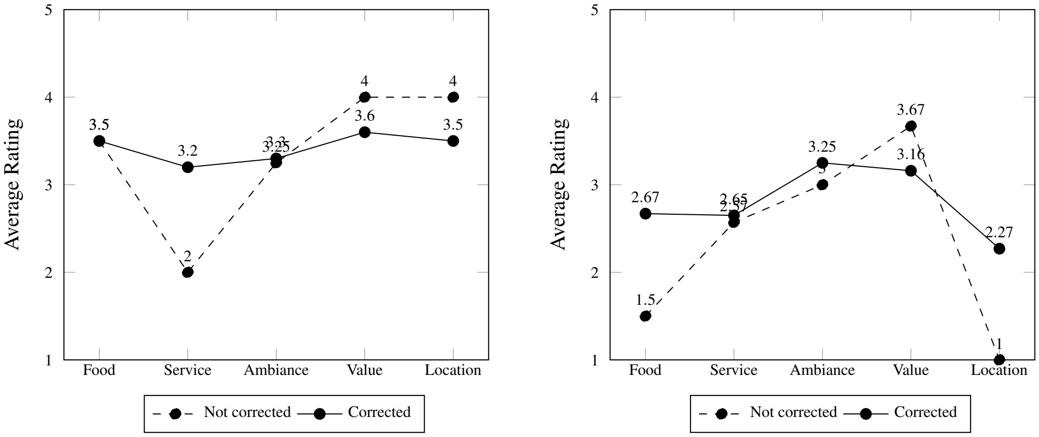
Table 6: Model Fit: Root Mean Squared Error (RMSE) across Imputations

Attribute	Our method	No Heterogeneity	Fixed Imputation Scores		
			Score 1	Score 3	Score 5
Overall	0.486	0.932	1.879	0.778	1.211
Food	0.488	0.881	2.631	0.973	1.323
Service	0.791	0.967	2.304	0.939	1.587
Ambiance	0.337	1.169	1.768	0.711	0.960
Value	0.438	0.924	1.534	0.706	1.132
Location	0.376	0.717	1.161	0.564	1.055

Table 7: Impact of Imputation on Attribute Ratings

	Average Correction		% of corrections ≥ 0.5	
	Chain	Independent	Chain	Independent
Food	0.50	0.13	45%	15%
Service	0.52	0.32	51%	32%
Ambiance	0.83	0.69	66%	61%
Price	1.09	0.81	76%	69%
Location	1.45	1.10	83%	76%
N: 2719				

Figure 4: Change in Average Attribute Rating



(a) Independent restaurant in Phoenix

(b) Chain restaurant in Las Vegas

Correction for Attribute Mention (Silence) in Attribute Ratings

We now illustrate how correcting for attribute mentions (silence) through imputation at the individual review level can impact overall attribute rating for a restaurant. In Table 7, we can see that correction for missing attributes has significant impact on attributes that are missed more frequently: value and location in general, and food for chain restaurants. The correction could be either upward or downward depending on attribute, restaurant type and reviewer type. For example, at an independent restaurant in Phoenix where most reviewers are found to remain silent about service at higher satisfaction levels, observed service ratings are lower than actual service ratings after imputing for missing attribute ratings. Then, correction results in higher service ratings than observed ratings (Figure 4a). Food and ambiance scores barely change, and value and location scores slightly go up after imputation for this restaurant. In Figure 4b, we illustrate a chain restaurant in Las Vegas where many of the reviewers miss food and location attributes, when satisfied, and miss value rating when dissatisfied. Here food and location scores go up and value score go down. Overall this shows that our imputation approach based on restaurant observables, rater observables and unobservable heterogeneity is extremely flexible in its imputations and the ability to correct for missing attribute ratings.

CONCLUSION

The paper addresses the general problem of using unstructured text data to generate quantifiable market feedback typically obtained through surveys; the specific application is to use restaurant reviews to generate attribute level ratings of restaurants. The paper addresses two novel and challenging problems around online text reviews: (i) convert text into *fine-grained numerical sentiment scores* on pre-specified attributes (e.g., food, service) by accounting for language structure; and (ii) accounting for *missing attributes* in attribute sentiment scoring. For the first problem, it uses a deep learning convolution-LSTM model that exploits the spatial and sequential structure of language to improve sentiment classification, especially on known types of "hard" sentences in NLP. For addressing missing attributes, the paper develops and estimates a structural model of reviewer rating behavior that takes into account the data generating process to develop a model-based imputation procedure to address attribute silence. Overall, the paper illustrates the value of combining "engineering" thinking underlying machine learning approaches with "social science" thinking from econometrics to answer novel marketing questions.

Substantively, the paper identified three segments of reviewers—the smallest but most active reviewers ("Status Seeking Regulars,") the largest segment ("Altruistic Mass,") who review without reward expectations and ("Emotive Irregulars,") who review infrequently, but write about attributes they are extremely satisfied or dissatisfied about. Our insights around attribute silence in reviews shows that informativeness and need to praise/vent drive more of the writing than the importance of the attribute. Not only does this contribute to the literature on why people engage in online word of mouth (Berger 2014), it also has implications for using reviews as a source of data for needs/benefits identification. In particular, contrary to conventional wisdom, the frequency of mentions of a benefit or a topic may not necessarily be a proxy of its importance for all types of reviewers.

We conclude with a discussion on some suggestions for future research. First, while our method improved performance accuracy for all hard sentence types, there is clearly more room for improvement. It would be useful to evaluate the performance of some of the recent transformer-based language models (e.g., BERT, GPT-2) that are contextual and use "attention mechanism" for our task of sentence-level, fine-grained sentiment analysis of hard sentences; as these models have improved the state-of-the-art in several language tasks especially involving longer sequences like paragraphs (though our context is only a sentence). Second, our paper shows that many traditional lexicon based approaches used for text analysis have significant levels of classification error even with respect to sentiment valence and more so with respect to fine-grained sentiment scoring. It would be useful to study whether the "measurement error" in the conversion from text to numeric data induced by these simpler (but intuitive methods) leads to attenuation bias that impact the substantive conclusions in social science research. Third, given our objective to summarize only reviews that have been written (as that is what impacts consumer perception of sentiments), we abstracted away from the issue of selection in the decision to write reviews. Further, we abstracted away from the issue of fake reviews/review shading. It would be worthwhile combining our content analysis at the attribute level with work on fake reviews/review shading to get a richer understanding of how to correct for issues of selection and fake reviews in tracking WOM.

REFERENCES

Archak N, Ghose A, Ipeirotis PG (2011) Deriving the pricing power of product features by mining consumer reviews. *Management science* 57(8):1485–1509.

Arcidiacono P, Jones JB (2003) Finite mixture distributions, sequential likelihood and the em algorithm. *Econometrica* 71(3):933–946.

Arcidiacono P, Miller RA (2011) Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica* 79(6):1823–1867.

Athey S, Bayati M, Doudchenko N, Imbens G, Khosravi K (2018) Matrix completion methods for causal panel data models. Technical report, National Bureau of Economic Research.

Bengio Y, CA M (2015) Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. *corr abs/1502.04390* .

Berger J (2014) Word of mouth and interpersonal communication: A review and directions for future research. *Journal of consumer psychology* 24(4):586–607.

Berger J, Sorensen AT, Rasmussen SJ (2010) Positive effects of negative publicity: When negative reviews increase sales. *Marketing Science* 29(5):815–827.

Bi JW, Liu Y, Fan ZP, Zhang J (2019) Wisdom of crowds: Conducting importance-performance analysis (ipa) through online reviews. *Tourism Management* 70:460–478.

Brahma S (2018) Improved sentence modeling using suffix bidirectional lstm. *arXiv preprint arXiv:1805.07340* .

Büschken J, Allenby GM (2016) Sentence-based text analysis for customer reviews. *Marketing Science* 35(6):953–975.

Büschken J, Allenby GM (2020) Improving text analysis using sentence conjunctions and punctuation. *Marketing Science* 39(4):727–742.

Chevalier JA, Mayzlin D (2006) The effect of word of mouth on sales: Online book reviews. *Journal of marketing research* 43(3):345–354.

Culotta A, Cutler J (2016) Mining brand perceptions from twitter social networks. *Marketing science* 35(3):343–362.

Dhar V, Chang EA (2009) Does chatter matter? the impact of user-generated content on music sales. *Journal of Interactive Marketing* 23(4):300–307.

Doshi-Velez F, Kim B (2017) Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* .

Duan W, Gu B, Whinston AB (2008) Do online reviews matter?—an empirical investigation of panel data. *Decision support systems* 45(4):1007–1016.

Dubois D, Chae I, Niessing J, Wee J (2016) Accorhotels and the digital transformation: Enriching experience trough content strategies along the customer journey. *INSEAD Business School* .

Feldman R (2013) Techniques and applications for sentiment analysis. *Communications of the ACM* 56(4):82–89.

Ganu G, Elhadad N, Marian A (2009) Beyond the stars: improving rating predictions using review text content. *WebDB*, volume 9, 1–6 (Citeseer).

Gentzkow M, Kelly B, Taddy M (2019) Text as data. *Journal of Economic Literature* 57(3):535–74.

Ghose A, Ipeirotis PG (2007) Designing novel review ranking systems: predicting the usefulness and impact of reviews. *Proceedings of the ninth international conference on Electronic commerce*, 303–310 (ACM).

Godes D, Mayzlin D (2004) Using online conversations to study word-of-mouth communication. *Marketing science* 23(4):545–560.

Goodfellow IJ, Bengio Y, Courville AC (2016) *Deep Learning*. Adaptive computation and machine learning (MIT Press).

Gurney N, Loewenstein G (2019) Filling in the blanks: What restaurant patrons assume about missing sanitation inspection grades. *Journal of Public Policy & Marketing* 0743915619875419.

Harris ZS (1954) Distributional structure. *Word* 10(2-3):146–162.

Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput.* 9(8):1735–1780, ISSN 0899-7667.

Hollenbeck B (2018) Online reputation mechanisms and the decreasing value of chain affiliation. *Journal of Marketing Research* 55(5):636–654.

Huang JL, Liu M, Bowling NA (2015) Insufficient effort responding: Examining an insidious confound in survey data. *Journal of Applied Psychology* 100(3):828.

Joachims T (2002) *Learning to classify text using support vector machines*, volume 668 (Springer Science & Business Media).

Johnson R, Zhang T (2017) Deep pyramid convolutional neural networks for text categorization. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Vol 1: Long Papers).*

Kim Y (2014) Convolutional neural networks for sentence classification. *arXiv:1408.5882* .

Krosnick JA (1991) Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied cognitive psychology* 5(3):213–236.

Le Mens G, Kovács B, Avrahami J, Kareev Y (2018) How endogenous crowd formation undermines the wisdom of the crowd in online ratings. *Psychological science* 29(9):1475–1490.

Lee TY, Bradlow ET (2011) Automated marketing research using online customer reviews. *Journal of Marketing Research* 48(5):881–894.

Li X, Hitt LM (2008) Self-selection and information role of online product reviews. *Information Systems Research* 19(4):456–474.

Li Y, Lu SF, Lu LX (2019) Do yelp reviews influence consumer choice in the presence of government ratings? evidence from us nursing homes. *Evidence from US Nursing Homes (October 1, 2019)* .

Little RJ, Rubin DB (2019) *Statistical analysis with missing data*, volume 793 (John Wiley & Sons).

Liu B, et al. (2010) Sentiment analysis and subjectivity. *Handbook of natural language processing* 2(2010):627–666.

Liu X, Lee D, Srinivasan K (2019) Large-scale cross-category analysis of consumer review content on sales conversion leveraging deep learning. *Journal of Marketing Research* 56(6):918–943.

Luca M (2016) Reviews, reputation, and revenue: The case of yelp. com .

Luca M, Vats S (2013) Digitizing doctor demand: The impact of online reviews on doctor choice. *Cambridge, MA: Harvard Business School* .

Luca M, Zervas G (2016) Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science* 62(12):3412–3427.

Mayzlin D, Dover Y, Chevalier J (2014) Promotional reviews: An empirical investigation of online review manipulation. *American Economic Review* 104(8):2421–55.

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Mittal V, Katrichis JM, Kumar P (2001) Attribute performance and customer satisfaction over time: evidence from two field studies. *Journal of Services Marketing* .

Mittal V, Kumar P, Tsiros M (1999) Attribute-level performance, satisfaction, and behavioral intentions over time: a consumption-system approach. *Journal of Marketing* 63(2):88–101.

Mohan K, Pearl J (2018) Graphical models for processing missing data. *arXiv preprint arXiv:1801.03583* .

Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. *Proceed-*

ings of the 27th International Conference on International Conference on Machine Learning, 807–814, ICML'10, ISBN 978-1-60558-907-7.

Netzer O, Feldman R, Goldenberg J, Fresko M (2012) Mine your own business: Market-structure surveillance through text mining. *Marketing Science* 31(3):521–543.

Nielsen FÅ (2011) A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903* .

Onishi H, Manchanda P (2012) Marketing activity, blogging and sales. *International Journal of Research in Marketing* 29(3):221–234.

Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. *LREc*, volume 10.

Peloza J, Ye C, Montford WJ (2015) When companies do good, are their products good for you? how corporate social responsibility creates a health halo. *Journal of Public Policy & Marketing* 34(1):19–31.

Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Puranam D, Narayan V, Kadiyali V (2017) The effect of calorie posting regulation on consumer opinion: a flexible latent dirichlet allocation model with informative priors. *Marketing Science* 36(5):726–746.

Rubin DB (1976) Inference and missing data. *Biometrika* 63(3):581–592.

Schouten K, Frasincar F (2015) Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28(3):813–830.

Slovic P, MacPhillamy D (1974) Dimensional commensurability and cue utilization in comparative judgment. *Organizational Behavior and Human Performance* 11(2):172–194.

Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2):267–307.

Timoshenko A, Hauser JR (2018) Identifying customer needs from user-generated content. *Marketing Science (Forthcoming)* .

Tirunillai S, Tellis GJ (2014) Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research* 51(4):463–479.

Wang H, Lu Y, Zhai C (2010) Latent aspect rating analysis on review text data: a rating regression approach. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 783–792 (ACM).

Wang J, Yu LC, Lai KR, Zhang X (2016) Dimensional sentiment analysis using a regional cnn-lstm model. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Vol 2: Short Papers)*, 225–230.

Xu X (2019) Examining the relevance of online customer textual reviews on hotels' product and service attributes. *Journal of Hospitality & Tourism Research* 43(1):141–163.

Zhou C, Sun C, Liu Z, Lau F (2015) A c-lstm neural network for text classification. *preprint arXiv:1511.08630* .

Zhu F, Zhang X (2010) Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. *Journal of marketing* 74(2):133–148.

APPENDIX

Estimation Algorithm: Detailed Steps

Step 0. The Initialization Step: We describe the detailed steps of the estimation algorithm for the structural model below.

Determine initial value of parameters $\Theta^{(0)}$ assuming no unobserved heterogeneity across reviewers, i.e., Set $G = 1$. To obtain $\Theta^{(0)}$ other than $\{q_i^{g(0)}\}$ for $g = 1, ..., G$, maximize the likelihood $\sum_i^m lnL_i^{g(0)}$ when $G = 1$. To do so, separately maximize each of the three component likelihoods in equation (8).

i. Maximize $\sum_i^m lnL_i^{Ag(0)}$ with only observations that have the attribute ratings reported, in order to obtain $\{\alpha_k^{(0)}, \beta_{ik}^{(0)}, C_{ks}^{g(0)}\}$.

ii. Use $\{\alpha_k^{(0)}, \beta_{ik}^{(0)}, C_{ks}^{g(0)}\}$ to determine values of X_{ijk}^w and maximize $\sum_i^m lnL_i^{Wg(0)}$ to obtain $\delta_k^{g(0)}$.

iii. Update $P_{ijks}^{g(0)}$ using $\{\alpha_k^{(0)}, \beta_{ik}^{(0)}, C_{ks}^{g(0)}, \delta_k^{g(0)}\}$ and Bayes rule.

iv. Iterate between (i)-(iii). When maximizing $\sum_i^m lnL_i^{Ag(0)}$, use both reported and missing attribute ratings given $P_{ijks}^{g(0)}$.

v. Maximize $\sum_i^m lnL_i^{Rg(0)}$ to obtain $\gamma_0^{g(0)}, \gamma_k^{g(0)}, \sigma^{g(0)}$.

Using the estimates above from the model without unobserved heterogeneity, initialize values of parameters for the model with unobserved heterogeneity with G segments as follows. For any segment $g \in G$, $\{C_{ks}^{g(0)}, \delta_k^{g(0)}, \gamma_0^{g(0)}, \gamma_k^{g(0)}, \sigma^{g(0)}\} = \{C_{ks}^{(0)}, \delta_k^{(0)}, \gamma_0^{(0)}, \gamma_k^{(0)}, \sigma^{(0)}\}$. Assume each segment is of the same size and set $\pi^{g(0)} = 1/G$, for all g.

Step 1. Segement Probability Update:

Given the parameters in the n^{th} iteration $\{\alpha_k^{(n)}, \beta_{ik}^{(n)}, C_{ks}^{g(n)}, \delta_k^{g(n)}, \gamma_0^{g(n)}, \gamma_k^{g(n)}, \sigma^{g(n)}\}$, for each reviewer $i = 1, 2, ..., m$, compute $q_i^{g(n+1)}$:

$$q_i^{g(n+1)} = \frac{\pi^{g(n)} L_i^{g(n)}}{\sum_g^G \pi^{g(n)} L_i^{g(n)}}$$

Then, update $\pi^{g(n+1)} = \frac{1}{N}\sum_i^N q_i^{g(n+1)}$.

Step 2. The Likelihood Maximization (Conditional on Segment Probability):

Estimate the parameters of the $(n+1)^{th}$ iteration $\Theta^{(n+1)}$, other than $q_i^{g(n+1)}$. Given the segment probability $q_i^{g(n+1)}$ in the E step, maximize the likelihood:

$$\sum_i^N ln\Big[\sum_g^G q_i^{g(n+1)} L_i^{g(n+1)}\Big].$$

As in step 0,

i. Start with reported attribute ratings to maximize $\sum_i^N ln\Big[\sum_g^G q_i^{g(n+1)} L_i^{Ag(n+1)}\Big]$.

ii. Maximize $\sum_i^N ln\Big[\sum_g^G q_i^{g(n+1)} L_i^{Wg(n+1)}\Big]$.

iii. Revise $P_{ijks}^{g(n+1)}$.

iv. Using both reported and missing attribute ratings, iterate between

v. Iterate between Steps (i)-(iii).

vi. Maximize $\sum_i^N ln\Big[\sum_g^G q_i^{g(n+1)} L_i^{Rg(n+1)}\Big]$.

Iterate between Step 1 and Step 2 steps until convergence.

ONLINE APPENDIX

OA.1 Experiment to show Attribute Ratings Improve Decision Making

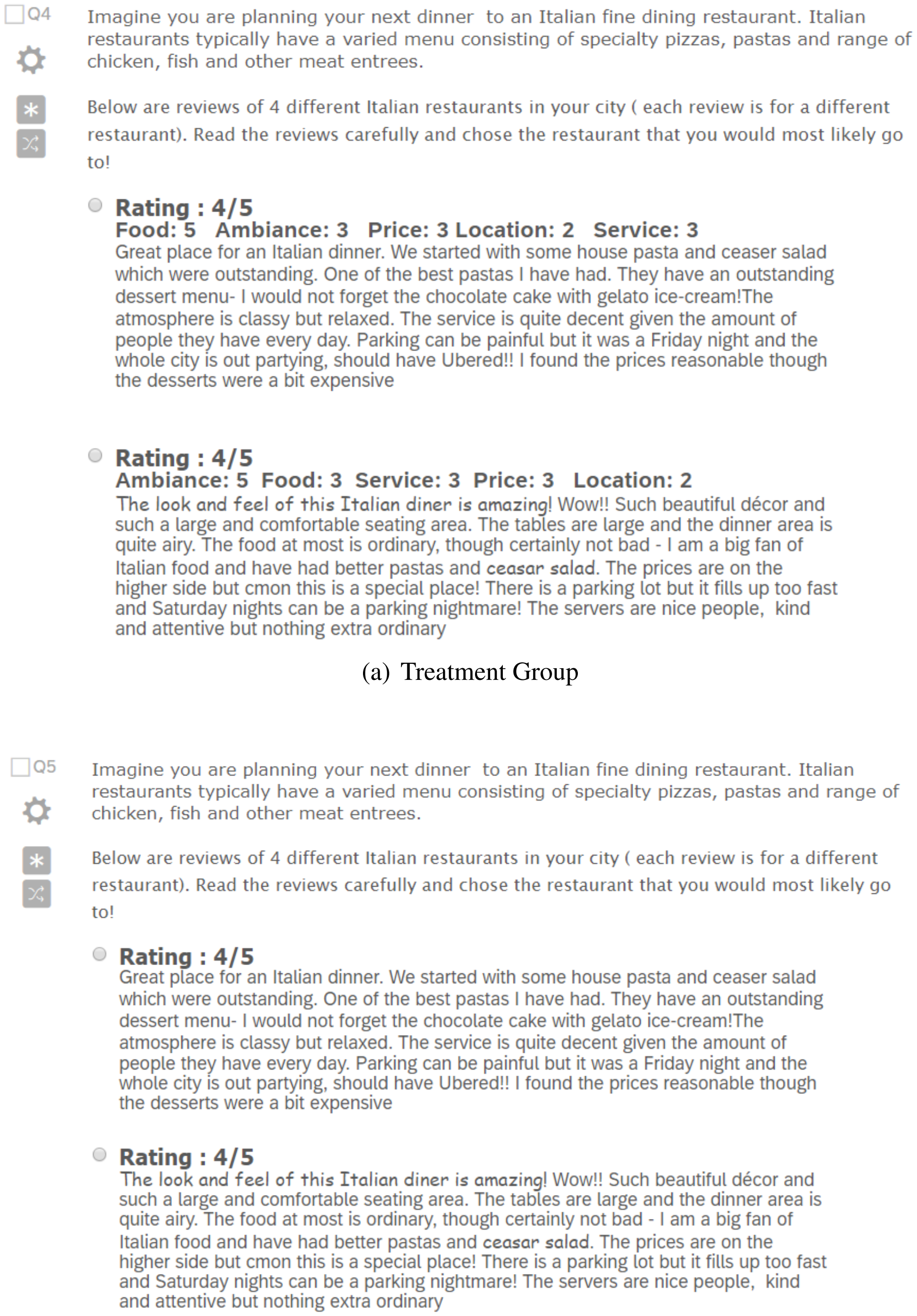
In this section, we describe the Amazon Mechanical Turk (Mturk) Experiment we run to motivate the importance of attribute level ratings. First, to establish that enhanced ratings are useful for customers to make better decisions, we conducted a 2×2 between subjects study on MTurk with 165 participants. Both the treatment and control groups are shown 4 restaurant reviews and asked to chose a restaurant. Every restaurant is extremely good at one of the attributes— food, service, price or ambiance and average on other attributes. The only additional information given to the treatment group is enhanced attribute level ratings. See Figure OA1 for details of the study design. We compare the treatment and control groups on two parameters— match and attention. We consider a match when a person's restaurant choice matches with their separately elicited preference i.e. a person who says she values food chooses the restaurant that has excellent food and so on. We get our measure of attention based on whether the survey respondent correctly answers the attention check question: "How many restaurant choices did you have in the previous question?" asked immediately after the restaurant choice question.

We show in Table OA1 that providing attribute sentiment scores in addition to text significantly improves the ability of customers to choose restaurants consistent with their separately elicited preferences over restaurant attributes. There is also a significant positive impact on attention. The fact that the treatment group is more attentive and makes choices more consistent with preferences shows that attribute level ratings reduce the cognitive burden of consumers and helps them in decision making.

Table OA1: Match and Attention Comparison: Treatment (Attribute Scores)

	N	Mean (SD)	
		Match	Attention
Treatment	74	0.7 (0.46)	0.94 (0.46)
Control	90	0.38 (0.49)	0.83(0.49)
		$p<0.01$	$p<0.05$

Figure OA1: Mturk experiment: Importance of attribute sentiments



□Q4

Imagine you are planning your next dinner to an Italian fine dining restaurant. Italian restaurants typically have a varied menu consisting of specialty pizzas, pastas and range of chicken, fish and other meat entrees.

Below are reviews of 4 different Italian restaurants in your city (each review is for a different restaurant). Read the reviews carefully and chose the restaurant that you would most likely go to!

○ **Rating : 4/5**
Food: 5   Ambiance: 3   Price: 3 Location: 2   Service: 3
Great place for an Italian dinner. We started with some house pasta and ceaser salad which were outstanding. One of the best pastas I have had. They have an outstanding dessert menu- I would not forget the chocolate cake with gelato ice-cream!The atmosphere is classy but relaxed. The service is quite decent given the amount of people they have every day. Parking can be painful but it was a Friday night and the whole city is out partying, should have Ubered!! I found the prices reasonable though the desserts were a bit expensive

○ **Rating : 4/5**
Ambiance: 5 Food: 3 Service: 3 Price: 3   Location: 2
The look and feel of this Italian diner is amazing! Wow!! Such beautiful décor and such a large and comfortable seating area. The tables are large and the dinner area is quite airy. The food at most is ordinary, though certainly not bad - I am a big fan of Italian food and have had better pastas and ceasar salad. The prices are on the higher side but cmon this is a special place! There is a parking lot but it fills up too fast and Saturday nights can be a parking nightmare! The servers are nice people, kind and attentive but nothing extra ordinary

(a) Treatment Group

□Q5

Imagine you are planning your next dinner to an Italian fine dining restaurant. Italian restaurants typically have a varied menu consisting of specialty pizzas, pastas and range of chicken, fish and other meat entrees.

Below are reviews of 4 different Italian restaurants in your city (each review is for a different restaurant). Read the reviews carefully and chose the restaurant that you would most likely go to!

○ **Rating : 4/5**
Great place for an Italian dinner. We started with some house pasta and ceaser salad which were outstanding. One of the best pastas I have had. They have an outstanding dessert menu- I would not forget the chocolate cake with gelato ice-cream!The atmosphere is classy but relaxed. The service is quite decent given the amount of people they have every day. Parking can be painful but it was a Friday night and the whole city is out partying, should have Ubered!! I found the prices reasonable though the desserts were a bit expensive

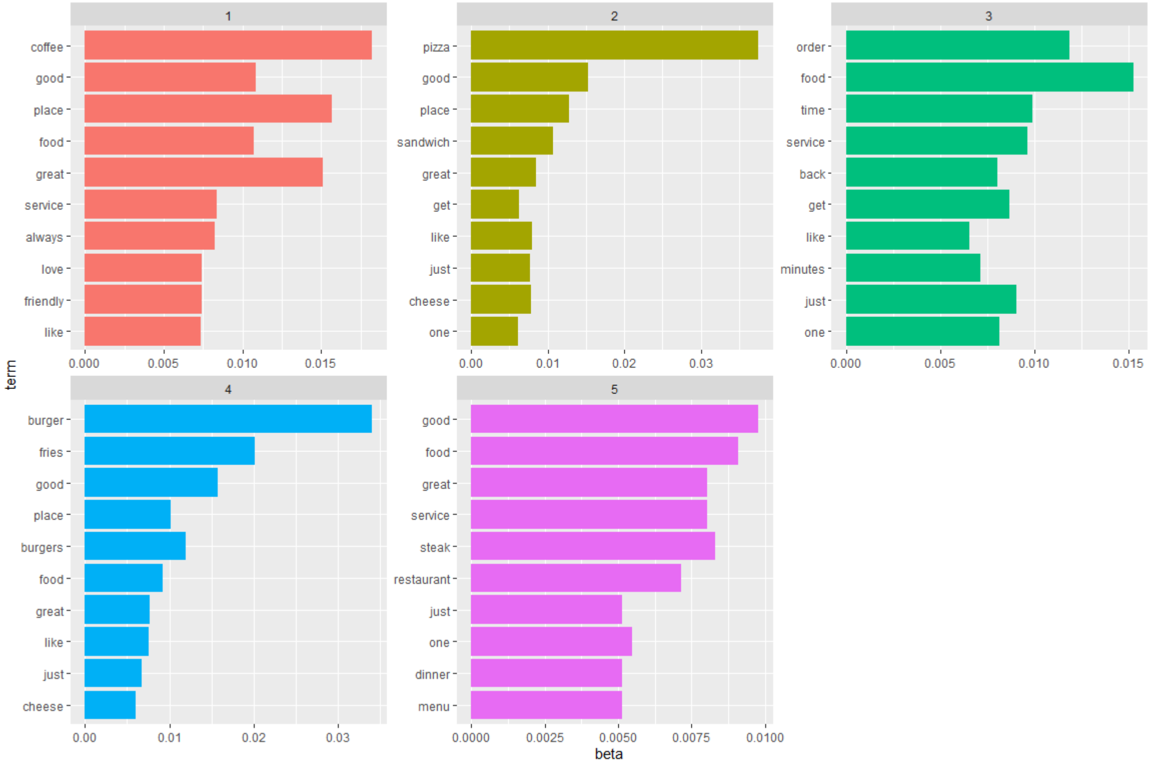
○ **Rating : 4/5**
The look and feel of this Italian diner is amazing! Wow!! Such beautiful décor and such a large and comfortable seating area. The tables are large and the dinner area is quite airy. The food at most is ordinary, though certainly not bad - I am a big fan of Italian food and have had better pastas and ceasar salad. The prices are on the higher side but cmon this is a special place! There is a parking lot but it fills up too fast and Saturday nights can be a parking nightmare! The servers are nice people, kind and attentive but nothing extra ordinary

(b) Control Group

OA.2 LDA and s-LDA for exploratory topic analysis

We use document level LDA and s-LDA as an exploratory tool to identify which topics are discussed in reviews. Figure OA2a and OA2b present the topics identified from the document level LDA and s-LDA. Table OA2 lists the important keywords characteristic of these topics.

Figure OA2a: LDA topics for Yelp review corpus (with seed words)



OA.3: Composition of Training and Test Data Sets

In Table OA3 we describe the composition of various training and test data sets used to ensure class balance. In OA4, we show the share of different types of sentences in our corpus.

OA.3 A Hybrid convolutional-LSTM Deep Learning Architecture

In this section, we include a more detailed discussion of the two most important layers of the hybrid CNN-LSTM: the convolutional layer and the long short term memory layer.

Table OA2: Top attribute and sentiment words

Attribute	**Attribute words**	**Positive Sentiment Words**	**Negative Sentiment Words**
Food	Food, chicken, beef, steak, appetizers, cheese, bacon, pork, taste, waffle, dish, shrimp, side, fries, menu, options, vegetarian, meat, gluten, salads, burger, mac, bread, cornbread, ingredients, egg, pancake, portions, brunch, lunch, dinner, breakfast, snack, potatoes, selection, entrée, dessert, maincourse, cake, brownie, ice cream, drink, water, alchol, nonalcoholic, tea, coffee, mocha , vodka, tequila, mocktail, beer, cocktails, cellar, glasses, wine, water	delicious, good, great, fresh, tasty, rich, hot, juicy, perfect, impressed, impressive, overwhelming, crispy, crunchy, warm, authentic, savory, amazing, real, nice, filling ,fantastic, quality, favorite, decent, enormous, special, fluffy, perfection, addicting, hearty, satisfactory, green, outstanding, yummy	not good, not the best, underwhelming, less, light, limited, stale, cold, not fresh, disappointing, awful, salty, off, soggy, unsatisfactory, bland, tasteless, cold, undercooked,watery
Service	Server, waiter, waitress, girl, boy, owner, ladies, manager, staff, bartender, customer service, service, seated, wait time, presentation, hostess, tip, chefs, front desk, reception, greeted, seated, filled, serve, refill, wait time	responsive, quick, friendly, accommodating, helpful, knowledgeable, fast, regular, great, immediately, amazing, kind, polite, great, smile, smiling, attentive, sweet	slow, bored, long, less, irritated, displeased, busy, inattentive, did not ask, rude, cold, long time, queue, long, angry, impolite, careless, dishonest, lied
Price	Price, dollars, money, numbers ($1, $ 5 etc.), credit, debit, cash, payment, discount, deal, offer, pay, total, charge, happy hour, save, spent, worth, bucks, cost, bill, tip, coupon	totally worth, cheap, good deal, bargain, free, worthy, inexpensive	expensive, pricy, pricey, steep, surchage, high, higher, overpriced, loot, too rich, lot, steep, additional charge
Location	location, located, street, address, spot, parking , college, office, airport, neighborhood, area, ny, vegas, california	near, nearby, convenient, walking, short, easy, safe, ample parking, on the way	far, secluded, away, shady, unsafe, dingy, long, travel time, no parking
Ambiance	atmosphere, ambience, ambiance, décor, decore, chair, sofa, tables, place, view, patio, terrace, washroom, restroom, design, furniture, crowd, casino, music, lounge, noise	Impressive, friendly, elegant, beautiful, cool, modern, upscale, outgoing, romantic, mind blowing classy, country ,inviting, big, spectacular, open, lively, very clean, nicely done, calm, positive vibe	busy, crowded, noisy, boring , loud, crunched, old, small, shabby, dirty, stinking, negative, wannabe, not great, shitty, dark, not airy

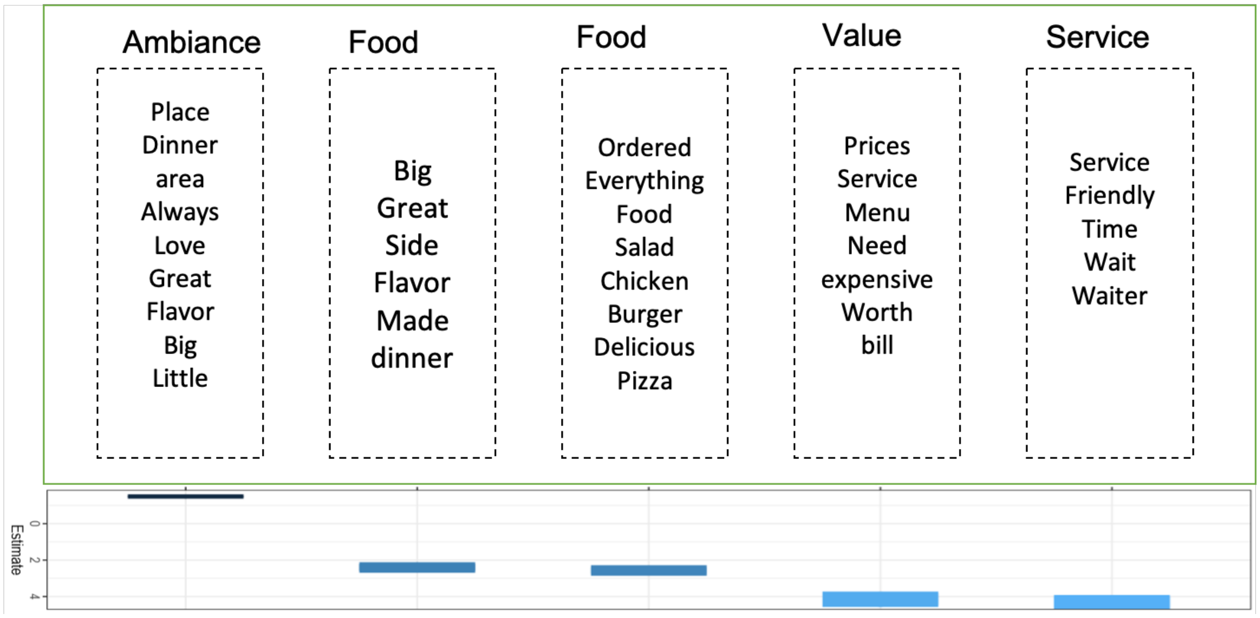
Table OA3: Class Balance: Attribute and Sentiment Classes (N: 2400)

Attribute			Sentiment		
Class	Training Data	Test Data	Class	Training Data	Test Data
Food	34%	37%	Negative	18%	25%
Service	21%	23%	Positive	35%	27%
Ambiance	14%	12%	Very Negative	11%	9%
Value	11%	10%	Very Positive	21%	27%
Location	4%	7%	Neutral	15%	11%

Table OA4: Distribution of Sentence Types (N: 706)

	Positive	Neutral	Negative	
Overall	52%	12%	36%	
Simple	64%	53%	34%	52%
Implied	6%	5%	32%	15%
Contrastive	7%	20%	11%	10%
Long	26%	24%	28%	27%

Figure OA2b: SLDA topics for Yelp review corpus



Convolution Layer. The first feature generating layer in our architecture that follows the embedding layer is the convolution layer. Convolution refers to a cross-correlation operation that captures the interactions between a variable sized input and a fixed size weight matrix called filter (Goodfellow et al. 2016). A convolutional layer is a collection of several filters where each filter is a weight matrix that extracts a particular feature of the data. In the context of text classification, a filter could be extracting features like bi-grams that stand for negation *e.g. not good* or unigrams that stand for a particular attribute e.g. chicken. The two key ideas in a convolutional neural network are weight-sharing and sparse connections. Weight-sharing means using the same filter to interact with different parts of the data and sparse connection refers to the fact that there are fewer links between the neurons in adjacent layers. These two features reduce the parameter space of the model to a great extent thereby lowering the training time and number of training examples needed. Thus, CNN-based models take relatively little time to train compared to fully-connected networks or sequential networks. Training a CNN involves fixing the weight matrix of the shared filters by repeatedly updating the weights with the objective of minimizing a loss function that captures how far the predicted classification of the model is from the true class of training data.

An embedded sentence vector of dimension $n \times d$ enters the convolution layer. Filters of height h (where filter height denotes length of n-gram captured) and width d act on the input vector to generate one feature map each. For illustration purposes, let us consider a filter matrix F of size $h \times d$ that moves across the entire range of the input I of size $n \times d$, convolving with a subset of the input of size $h \times d$ to generate a feature map M of dimension $(n-h+1) \times 1$. A typical convolution operation involves computing a map by element-wise multiplication of a window of word vectors with the filter matrix in the following manner:

$$M(i,1) = \sum_{i=1}^{n-h+1} \sum_{m=1}^{h} \sum_{n=1}^{d} I(i+(m-1),n)F(m,n) \tag{9}$$

When there is a combination of filters of varying heights (say 1,2,3 etc.), we get feature maps of variable sizes ($n, n-1, n-2$ and so on).

Max-pooling and flattening operations are performed to concatenate variable size feature maps into a single feature vector that is passed to the next feature generating layer.

The role of the convolutional layer in this model is to extract phrase-level location invariant features that can aid in attribute and sentiment classification. A feature map emerging from a convolution of word vectors can be visualized as several higher-order representations of the original sentence like n-grams that capture negation like “not good” or “not that great experience” or n-grams that describe an attribute like “waiting staff” or “owner’s wife.” The number of filters to be used, N_f is fixed during hyper parameter tuning. Feature maps from all filters are passed through a non-linear activation function a_f with a small bias or constant term b to generate an output that would serve as input for the next stages of the model.

$$O_i = a_f(M_i + b) \tag{10}$$

The function f here can be any non-linear transformation that acts on the element-wise multiplication of the filter weights and word vectors plus a small bias term b. We use Rectified Linear Units (RELU) that is more robust in ensuring the network continues to learn for longer time periods compared to other activation functions like the tanh function (Nair and Hinton 2010). This activation function has the following format:

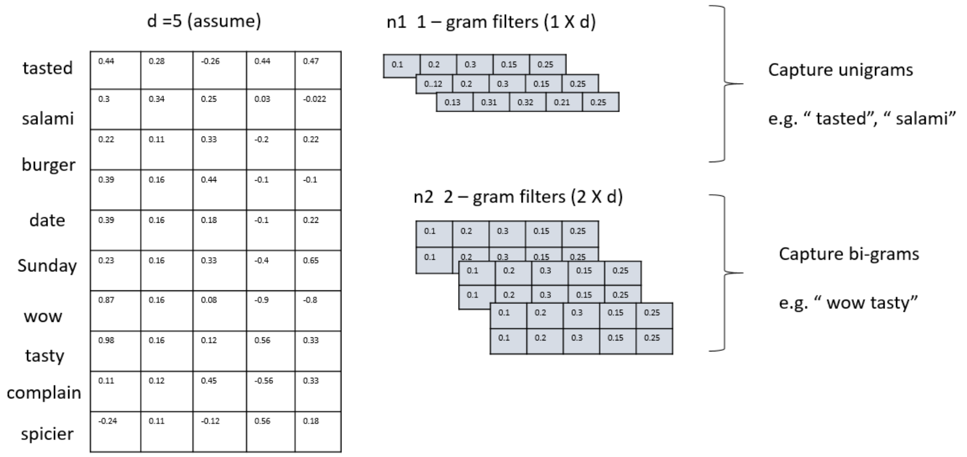
$$RELU(x) = max(0,x) \tag{11}$$

This activation function sets all negative terms in the feature maps to zero while preserving the positive outputs.

Figures OA3a and OA3b show the structure of the convolution layer and the convolution operation respectively. Figure OA3c shows a sample visualization of a feature map. During the course of training, each filter specializes in identifying a particular class. For instance, this filter has specialized in detecting *good food*.

Long Short Term Memory (LSTM) layer. The concatenated feature maps from the convolution layer are next fed into a Long Short Term Memory (LSTM) layer. LSTM is a special variant of the recurrent neural networks (RNN) that specialize in handling long-range dependencies. RNNs have a sequential structure and hence they can model inter-dependencies between the current input and the previous inputs using a history variable that is passed from one time period to the next. However, in practice, RNNs fail to do text classification tasks better than CNNs due to the "vanishing gradient" problem which causes a network to totally stop learning after some iterations (Nair and Hinton 2010). Vanishing gradients in the earlier layers of a recurrent neural network mainly result from a combination of non-linear activation functions like sigmoid and small weights in the later layers. LSTMs solve this problem by using a special memory unit with a fixed weight self-connection and linear activation function that ensures a constant non-vanishing error flow within the cell. Further, to ensure that irrelevant units do not perturb this cell, they employ a combination of gate structures that constantly make choices about what parts of the history need to be forgotten and what needs to be retained to improve the accuracy of the task at hand (Hochreiter and Schmidhuber 1997). This architecture has shown remarkable success in several natural language processing tasks like machine translation and speech to text transcription.

Figure OA3a: Convolutional Neural Network: filters and hyper parameters


d =5 (assume)
n1 1 – gram filters (1 X d)
Capture unigrams
e.g. “ tasted”, “ salami”
n2 2 – gram filters (2 X d)
Capture bi-grams
e.g. “ wow tasty”

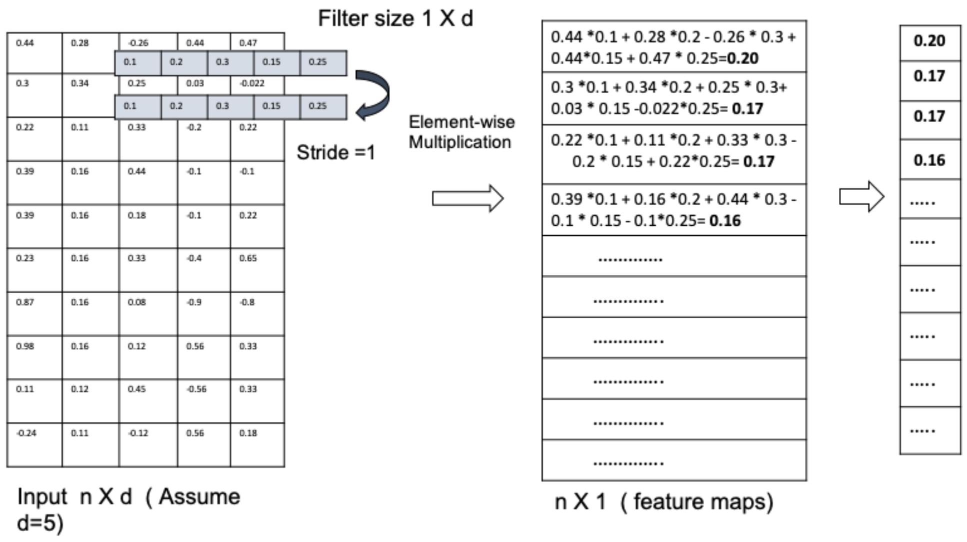
tasted	0.44	0.28	-0.26	0.44	0.47
salami	0.3	0.34	0.25	0.03	-0.022
burger	0.22	0.11	0.33	-0.2	0.22
	0.39	0.16	0.44	-0.1	-0.1
date	0.39	0.16	0.18	-0.1	0.22
Sunday	0.23	0.16	0.33	-0.4	0.65
wow	0.87	0.16	0.08	-0.9	-0.8
tasty	0.98	0.16	0.12	0.56	0.33
complain	0.11	0.12	0.45	-0.56	0.33
spicier	-0.24	0.11	-0.12	0.56	0.18

0.1	0.2	0.3	0.15	0.25
0..12	0.2	0.3	0.15	0.25
0.13	0.31	0.32	0.21	0.25

0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25

- The filter sizes (1,2,3), number of filters (n1,n2) and embedding dimension d are important tunable hyper parameters

Figure OA3b: Convolution Operation

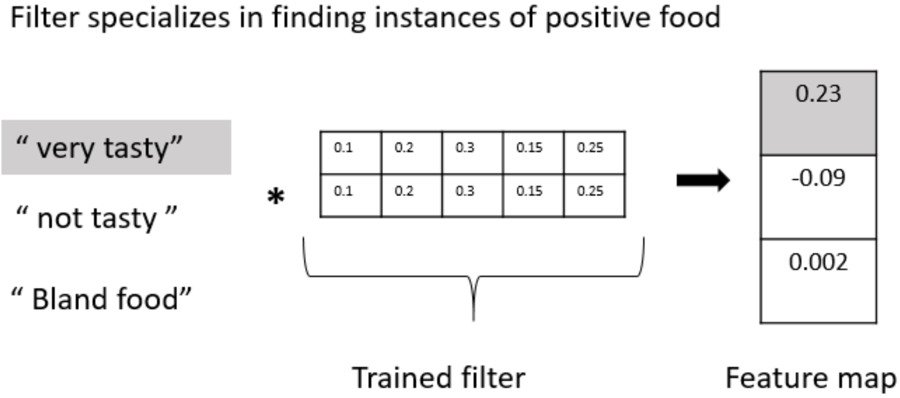


0.44	0.28	-0.26	0.44	0.47
0.3	0.34	0.25	0.03	-0.022
0.22	0.11	0.33	-0.2	0.22
0.39	0.16	0.44	-0.1	-0.1
0.39	0.16	0.18	-0.1	0.22
0.23	0.16	0.33	-0.4	0.65
0.87	0.16	0.08	-0.9	-0.8
0.98	0.16	0.12	0.56	0.33
0.11	0.12	0.45	-0.56	0.33
-0.24	0.11	-0.12	0.56	0.18

0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25

0.44 *0.1 + 0.28 *0.2 - 0.26 * 0.3 + 0.44*0.15 + 0.47 * 0.25=**0.20**	**0.20**
0.3 *0.1 + 0.34 *0.2 + 0.25 * 0.3+ 0.03 * 0.15 -0.022*0.25= **0.17**	**0.17**
0.22 *0.1 + 0.11 *0.2 + 0.33 * 0.3 - 0.2 * 0.15 + 0.22*0.25= **0.17**	**0.17**
0.39 *0.1 + 0.16 *0.2 + 0.44 * 0.3 - 0.1 * 0.15 - 0.1*0.25= **0.16**	**0.16**
....	.....
....	.....
....	.....
....	.....
....	.....
....	.....

Figure OA3c: Visualization of a Feature Map

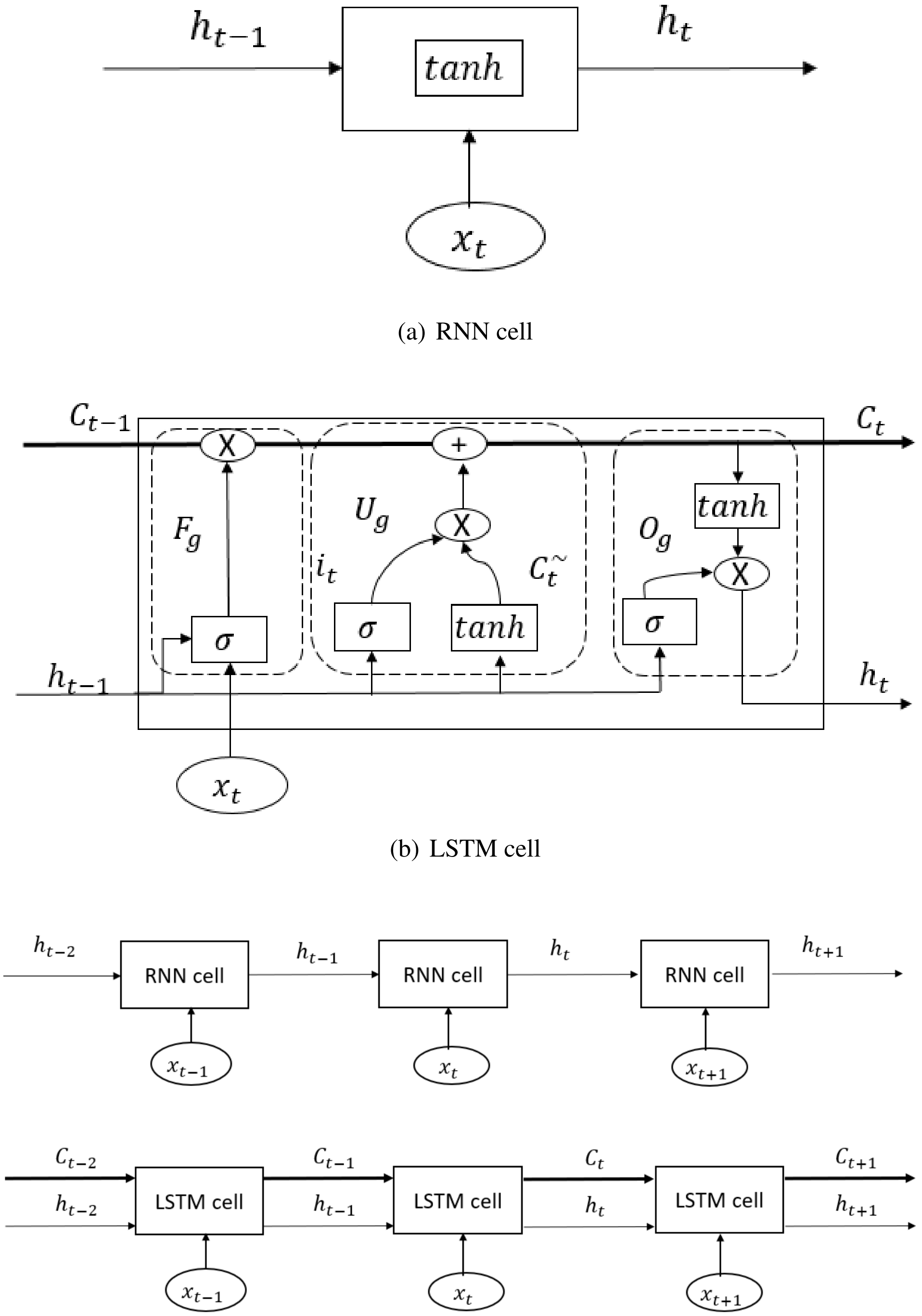


0.1	0.2	0.3	0.15	0.25
0.1	0.2	0.3	0.15	0.25

Figure OA4 is a comparison of RNN and LSTM architectures. In an RNN, the output at a particular time t is fed back into the same network in a feedback loop. In this way, a new input x_t interacts with the old history variable h_{t-1} to create the new output o_t and the a new history variable h_t. This is like in a relay race where each cell of the network passes on information of its past state to the next cell (but each cell is identical, and therefore it is equivalent to passing on the information to itself). The Long Short Term Memory (LSTM) cell differs from the RNN cell on two important aspects—the existence of a cell state C_t (the long term memory) and a combination of gates that regulate the flow of information into the cell state. The cell state is like a conveyor belt that stores the information that the network decides to take forward at any point in time t. Gates are sigmoidal units whose value is multiplied with the values of the other nodes. If the gate has a value of zero, it can completely block the information coming from another node whereas if the gate has a value $\in (0,1)$, it can selectively allow some portion of the information to pass. Thus, gates are like "regulators" of what information flows into and remains active within the system. The LSTM has three gates — a forget gate G_F, an update gate G_U and an output gate G_O.

Suppose x_t represents the input to the LSTM at a particular time t and h_{t-1} denotes the hidden state (or history) that is stored from a previous time period. At the first stage, the forget gate decides what part of the previous state needs to be forgotten or removed from the cell state. For instance, in a long sentence, once the LSTM has figured out that the sentence is primarily about the taste of

Figure OA4: Comparison of RNN and LSTM cells



(a) RNN cell

(b) LSTM cell

(c) Unrolled RNN and LSTM networks

a burger, it might chose to remove useless information regarding weather or day of the week that says nothing about food taste. The transition function for the forget gate can be represented as :

$$f_t = \sigma(W_f\dot{[}h_{t-1}, x_t] + b_f) \tag{12}$$

This equation is a typical neural network equation that involves an element-wise multiplication of a weight function with the hidden state h_{t-1} and current input x_t followed by the addition of a bias term and subsequent non-linearity. The other transition functions of the LSTM include an update function and an output function. The update function decides what part of the current input needs to be updated to the cell state. The output function first determines the output o_t for the current time period and subsequently, the new hidden state h_t that is passed to the next time period by selectively combining the current output and cell state contents that seem most relevant.

$$\begin{matrix}(13)\\(14)\\(15)\\(16)\\(17)\end{matrix}\qquad\begin{cases} i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \\ C_t = (f_t C_{t-1} + i_t \tilde{C}_t) \\ o_t = \sigma(W_o[h_t - 1, x_t] + b_o) \\ h_t = o_t \tanh(C_t) \end{cases}$$

All the weight matrices W_f, W_i, W_c and W_o are shared across different time steps. Thus, training an LSTM basically involves training these shared weight matrices by optimizing over a loss function.

OA.4 Additional Performance Metrics

In this section, we describe some additional performance metrics that were excluded in the main text for brevity. As we mentioned in the section *Performance Measures*, though accuracy is a first-order metric for hard problems like granular sentiment detection, we need other measures to refine model choice; especially among models with similar accuracy scores. These include some objective metrics like polarity preservation, precision and recall (derived from the confusion matrices) and some practical considerations like model building effort, scalability and interpretability. We now elaborate on each of these metrics and report the model comparison results.

Simple Confusion Matrix for Attribute Classification Accuracy: This confusion matrix helps to evaluate class-wise accuracy—doing so allows us to assess whether overall higher accuracy comes only from superior performance in high high-occurrence classes like food or a class like location that has few attribute words. We can assess whether the model is able to capture more complex classes like ambiance and service which manifest with a varied set of attribute words.

Polarity Reversal Confusion Matrix for Sentiment Accuracy: Though the CS literature typically uses accuracy as a performance metric for the fine-grained (multi-class) sentiment classification (Socher et al. 2013, Kim 2014), there can be other useful metrics of performance. For example, it may be useful to construct a polarity based coarse class: positive, neutral, negative and assess accuracy on the coarse classes classification because confusing sentiment class 1 with 4 or 5 (a polarity reversal) is worse than confusing 1 with 2 (same polarity). With this thought, we construct and report a polarity reversal confusion matrix for the models that have the best overall accuracy.

Table OA4b shows that the convolutional-LSTM model using Glove pre-trained embedding is slightly better than the one using self-trained embedding (though the overall accuracy is higher for the latter) because it preserves polarity better i.e. it mostly mis-classifies within the granular sentiment classes (positive, negative, neutral) and thus has lower polarity reversal. Table OA4c assesses attribute classification accuracy. We find that both the convolutional-LSTM based attribute classifiers using GloVe and self-trained embeddings do a fairly good job in classifying attributes across classes. Further, their performance is not driven simply by getting high-frequency classes like food right.

Also, refer Tables OA4d and OA4e to see how different models perform on metrics like precision, recall and F1 score.

Model Building Time Lexicon models take approximately 175-180 hours of construction time.

Table OA4b: Polarity Reversal Confusion Matrix (Sentiment Analysis)

	CNN			Convolutional-LSTM (self trained)			Convolutional-LSTM(Glove 300)		
True Class	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive
Very Negative	**31%**	7%	51%	**47%**	6%	47%	**71%**	2%	24%
Negative	**33%**	11%	63%	**45%**	6%	49%	**64%**	4%	35%
Neutral	14%	37%	49%	16%	33%	51%	44%	18%	39%
Positive	14%	9%	**77%**	15%	6%	**80%**	31%	5%	**64%**
Very Positive	9%	10%	**81%**	21%	2%	**88%**	18%	5%	**77%**

Table OA4c: Simple Confusion Matrix (Attribute Analysis)

	Convolutional-LSTM (self-trained)					Convolutional-LSTM (Glove 100)				
Predicted \ True	food	service	ambiance	value	location	food	service	ambiance	value	location
Food	**79%**	4%	2%	3%	2%	**75%**	6%	6%	3%	1%
Service	10%	**60%**	9%	5%	3%	7%	**76%**	8%	2%	0
Ambiance	8%	0	**58%**	3%	10%	2%	3%	**77%**	2%	2%
Value	10%	2	6%	**75%**	2%	8%	8%	6%	**74%**	4%
Location	8%	6%	11%	3%	**56%**	6%	14%	36%	3%	31%

Most of the time is spent on human-tagging of the 8575 attribute and sentiment words into specific classes using Amazon's Mechanical Turk. Similarly, the creation of training and test data sets for the supervised learning algorithms takes approximately 100 hours.[28] However, once created, we could use the same dataset to train and test a variety of machine learning and deep learning classifiers (e.g., SVM, Random Forest, Naive Bayes, CNN, LSTM and CNN-LSTM). After generating the training data, supervised learning models (including the deep learning models) need time for hyper parameter tuning and model training. Though this is an iterative process, all deep learning models take less than 10 minutes (in a quad core processor) for completing one training cycle and hence model calibration can be completed in 6-7 hours. Thus, model building is time-consuming for all algorithms but is a one-time activity.

[28] A human tagger takes around 1 minute to classify every word and 2-3 minutes to classify full sentences

Table OA4d: Precision and Recall (Sentiment Classification)

	CNN			CNN-LSTM (self-trained)			CNN-LSTM (Glove 300)		
Class	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Very Negative	38%	22%	28%	43%	24%	31%	30%	28%	29%
Negative	51%	27%	35%	52%	38%	44%	44%	59%	51%
Neutral	34%	37%	35%	46%	33%	39%	36%	18%	24%
Positive	34%	60%	43%	37%	68%	48%	40%	63%	49%
Very Positive	52%	40%	45%	67%	44%	53%	80%	58%	67%

Table OA4e: Precision and Recall (Attribute Analysis)

	CNN-LSTM (self trained)			CNN-LSTM (Glove 100)		
Class	Precision	Recall	F1	Precision	Recall	F1
ambiance	60%	58%	59%	55%	77%	64%
food	83%	79%	81%	86%	75%	80%
location	57%	56%	56%	73%	31%	43%
service	80%	60%	69%	71%	76%	73%
price	72%	75%	74%	76%	75%	76%

Scalability The more time-sensitive metric is scalability i.e. the time required for a trained model to classify new examples. With respect to the scalability metric, the deep learning classifiers clearly outperform the lexicon based classifiers with the machine learning classifiers in between the other two. The main reason is the "look-up" method employed by lexicon based methods. Every word in a sentence needs to be sequentially searched through the entire lexicon to determine its class. Hence, the lexicon methods need several hours to classify our corpus of 27,332 reviews comprising of 999,885 sentences. On the other hand, deep learning models are able to classify our entire review dataset comprising in approximately 18- 20 minutes.

Interpretability refers to how well a machine classifier can explain the reasoning or logic behind its classifications (Doshi-Velez and Kim 2017). In general, text mining methods differ in their strengths and weakness across various dimensions, there is no one method that is superior in all dimensions. Though the CNN-LSTM model outperforms all the other models in accuracy and scalability, however, it falls short in terms of interpretability with respect to lexicon methods.

OA.5 Sensitivity to Word Embeddings

Deep Learning models can be quite sensitive to several hyper-parameter tuning. Some of the important hyperparameters are embedding algorithm, filter sizes as well as the data used to train the embeddings. In Table OA5a, we show how accuracy is sensitive to the choice of word embedding algorithm as well as filter sizes. Table OA6 shows the difference in the type of vocabulary generated (in terms of similar words) when we use a pre-trained model versus a model trained from scratch on our data. The self-trained model's vocabulary is more relevant to the restaurant domain.

Table OA5a: Sensitivity to hyper parameter tuning (CNN-LSTM)

Hyper parameter	Configuration	Attribute Accuracy	Sentiment Accuracy
Embedding dimension	word2vec	58%	40%
	GloVe 100	68%	45%
	GloVe 300	66%	47%
Filter size	unigram	68%	40%
	bigram	67%	42%
	trigram	64%	38%
	[1,2]	66%	41%
	[1,2,3]	66%	42%
	[1,2,3,4]	66%	44%
	[1,2,3,4,5]	64%	47%

Table OA6: Top Similar Words in Word Embeddings: Cosine Similarity Scores

Attribute	**Self Trained**	**Glove Pre Trained**
Food	sushi (0.59), cuisine (0.57), meal (0.56), pizza (0.53), restaurant (0.49), dimsum (0.48), foods (0.48), fare (0.47), burgers (0.47), grub (0.46), salsa (0.44), menu (0.35), fish (0.33)	foods (0.66), eat (0.59), meat (0.56), meal (0.57), vegetables (0.54), nutrition (0.54), foodstuffs (0.53), cooking (0.52), bread (0.51), drinks (0.51), chicken (0.43), seafood (0.48))
Service	service (0.65), waitstaff (0.56), communication (0.55), staff (0.55), hospitality (0.48), consistently (0.48), experience (0.46), attitude (0.44), server (0.34), waiter (0.3)	service (0.79), news (0.48), phone (0.47), mail (0.47), provider (0.47), employee (0.46), customers (0.46), operate (0.45), serve (0.45)
Ambiance	ambience (0.95), atmosphere (0.91), decor (0.85), décor (0.75), vibe (0.75), environment (0.73), decoration (0.65), interior (0.63), aesthetic (0.62), cosy (0.61), setting (0.61)	ambience (0.85), décor (0.57), homey (0.58), convivial (0.53), rustic (0.53), woodsy (0.46), elegant (0.45), clubby (0.45), vibrant (0.44), opulent (0.43), surroundings (0.43), spacious (0.43), cozy (0.43), atmosphere (0.42)
Value	pricing (0.80), prices (0.75), value (0.60), rates (0.55), cost (0.54), markup (0.51), size (0.50), quality (0.50), pricey (0.46), overpriced (0.41), bucks (0.41), expensive (0.4)	prices (0.81), cost (0.61), value (0.57), pricing (0.55), share (0.54), premium (0.54), rates (0.53), inflation (0.52), sales (0.51), buy (0.5), dollar (0.5), low (0.5), rise (0.5)
Location	place (0.70), store (0.62), venue (0.61), starbucks (0.6), theatre (0.59), locale (0.58), intersection (0.58), marketplace (0.58), hotel (0.57), safeway (0.54), parking (0.3)	proximity (0.57), site (0.57), located (0.55), area (0.54), vicinity (0.54), venue (0.52), places (0.52), adjacent (0.5), nearby (0.49), geographical (0.48), situated (0.47), convenient (0.45), remote (0.44), facility (0.43)