ESSAYS ON FAKE REVIEW DETECTION, MANAGERIAL RESPONSE, AND CONSUMER PERCEPTIONS

by

Long Chen

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Management Science

at

The University of Wisconsin-Milwaukee

August 2021

ABSTRACT

ESSAYS ON FAKE REVIEW DETECTION, MANAGERIAL RESPONSE, AND CONSUMER PERCEPTIONS

by

Chen, Long

The University of Wisconsin-Milwaukee, 2021
Under the Supervision of Professor Sanjoy Ghose and Amit Bhatnagar

This dissertation investigates how online reviews and managerial responses jointly affect consumer perceptions. I first examine and compare the outcomes of multiple fake review classifiers using various algorithms, including traditional machine learning methods and recently developed deep learning methods (essay I). Then, based on the findings of the first essay, I examine the interrelationship between fake review detection, managerial response, and hotel ratings and ratings' growths (essay II).

The first essay is a comparative study on the methodology of identifying fake reviews. Although online reviews have attracted much attention from academia and industry for over fifteen years, how to identify fake reviews is still under study. In terms of methods, traditional machine learning classification methods were in dominant use. Recently, with the rise of deep learning methods in text analysis since the 2010s, researchers began applying new deep learning classification methods. In terms of features, the way to extract information from review content has been developing as the Natural Language Processing (NLP) area has made much progress since 2013. After that, researchers tried to apply both

deep learning algorithms and extract dense text features to build alternative systems for identifying fake reviews. Among various algorithms and features, how to choose and set up a good fake review detector, demands researchers to explore further to arrive at a widely accepted answer. This study is the first that applies both traditional machine learning and deep learning methods and compares across multiple datasets that vary in size, origin, and class distribution. This paper reports three findings. First, with new deep learning algorithms, classifiers perform better than classifiers using traditional machine learning methods in most cases, with only a few exceptions. Second, with dense word embeddings, classifiers perform better than classifiers using one-hot text features. Third, incorporating other numerical features boosts classification performance.

The purpose of the second study is twofold. First, to explore factors contributing to the likelihood of a review to receive a managerial response (MR), testing the impact of the fake review detection results, review congruency, review deviation, and the moderating role of hotel class. Second, to examine the association among online reviews, managerial responses, and hotel rating (growth rate), including both text similarity and fake review detection results as independent variables. This study is one of the first that introduces fake review detection and text similarity into research about MR, adding to the literature of MR in the context of Tripadvisor.com. Our findings indicate the following practical implications. (1) A truthful, detailed, and congruent review is more likely to receive an MR; (2) The percentage of truthful reviews has a strong and positive association with hotel rating and its growth. In an extreme situation, the hotel rating will go up by 0.21, and the rating growth rate will increase by 8.5% due to 100% truthful reviews; (3) Hotels should carefully choose which review(er)to respond to and make responses concise and matching while actively responding to reviews.

To

us three

and us all

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter/Essay I

Fake Online Review Detection Classifiers: A Comparative Study

Abstract

Online reviews have attracted much attention from academia and industry for over fifteen years. Many publications have emerged and offered answers to research questions. But the question about how to identify fake reviews is still under study. In terms of methods, traditional machine learning classification methods were in dominant use. Recently, with the rise of deep learning methods in text analysis in the 2010s, researchers began applying new deep learning classification methods. In terms of features, the way to extract information from review content has been developing as the Natural Language Processing (NLP) area has made much progress since 2013. After that, researchers tried to apply both deep learning algorithms and extract dense text features to build alternative systems for identifying fake reviews. This comparative study is one of the first to apply both machine learning and deep learning methods and to compare across multiple datasets that vary in size, origin, and class distribution. This paper reports three findings. First, with new deep learning algorithms, classifiers perform better than classifiers using traditional machine learning methods in most cases, with only a few exceptions. Second, with dense word embeddings, classifiers perform better than classifiers using one-hot text features. Third, incorporating other numerical features boosts classification performance. Contributions, limitations, and future research conclude the study.

Keywords: fake review detection, classifier, machine learning, deep learning, word

embeddings, Yelp

1 Introduction

Online reviews are statements that express someone's experiences, feelings, opinions, or suggestions about a product or service online, constituting a new type of word-of-mouth (WOM) information. A lot of research has been done on the impact of online reviews and found reviews are playing an increasingly important role to help the existing and potential consumers develop their thoughts and decisions, aid marketers in communication with their customers, and help a business grow and improve.

But, with the benefits from online reviews come drawbacks. In February 2004, because of a weeklong glitch, Amazon.ca unintentionally revealed the identities of 'anonymous' reviewers, briefly unmasking considerable self-reviewing by book authors(Harmon, 2004). Since then, studies on review spamming from economics, marketing, computer science, and other fields began to catch up. Researchers and practitioners point out that, for various reasons, a large number of online reviews are not made by valid customers but manufactured or manipulated by various entities, such as the manufacturer, competitors, and online reputation management companies that are paid for illegally offering fabricated, not truthful, reviews (Luca & Zervas, 2016). Businesses proactively using fake reviews do so mainly out of economic considerations. It is assumed that buying fake positive reviews can boost a business's rating, and it has shown that a one-star rating increase on Yelp, a major review platform, translated to an increase of 5% to 9% in revenues for a restaurant (Luca, 2016) and a hotel rated one star higher on TripAdvisor, Expedia, and Hotels.com on average has 27.8% higher demand (Lewis & Zervas, 2019).

When reading online reviews, people sometimes just really never know which review or

reviewer to trust because the widespread presence of fake reviews spoils the trustworthiness of reviews. Ideally, untruthful reviews that broadcast false information and skew the attention of viewers should be banned or rejected in the first place, or at least differentiated and then removed. This would leave only truthful online reviews that are informative and useful to customers and marketers. An automatic, reliable classification model of fake review detection is a way to defend a healthy e-commerce environment (Barbado et al., 2019).

Jindal and Liu (Jindal & Liu, 2008) conduct the first study on review spam in 2008. The study shows opinion spam in reviews is widespread and presents novel techniques to detect duplicate fake reviews. After that, studies on review manipulation attracted significant attention from areas of computer science and marketing (Jindal & Liu, 2008; Rayana & Akoglu, 2015). Studies have shown that the intention of rigged boosting/belittling the reputation of producers/competitors is one major motive of distributing untruthful positive/negative reviews, which make the rating higher/lower than it should be. The study of Mayzlin et al. shows the relationship between ownership structure and review manipulation (Mayzlin et al., 2014).

In the aspect of fake review detection techniques, many studies in computer science show building up an automatic classification model is promising. The classification methods ever used mainly concentrate on traditional statistical methods such as Logistic Regression, and classical machine learning (ML) algorithms. As for the part of features, in past studies, psycholinguistic features and one-hot text features like word counts, and other numerical features that are derived from the review, reviewer, and product information, have been used to analyze. Around 2013, breakthroughs happened to the natural language processing (NLP) area, which is a multidisciplinary area focusing, in particular, on how to program computers to

process and analyze unstructured natural language data. New alternative methods like representation learning and deep learning (DL), especially neural network-style methods, emerged to solve many NLP tasks including spam review detection. After that, researchers gradually began to adopt DL algorithms, such as Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN), and include features like n-gram, word2vec to train the model. These DL algorithms aim to build up more modern models, which have good potential to get a better performance like higher accuracy (Jain et al., 2019; Shahariar et al., 2019).

But comparative research is rare, which compares the classifiers using traditional ML methods and DL methods. Although plenty of papers claim that their proposed models identify fake reviews well, a few important questions remain open to study. Two research questions drive this study. First, do DL algorithms absolutely perform better than ML algorithms? Second, which approach to extract text features is better to enhance classification accuracy, word embeddings, or word counts?

This study aims to find insights into the impact of DL approaches on model performance by exploring various combinations of model approaches and feature sets across multiple datasets. The main findings of this paper are three-fold. First, with new DL algorithms, classifiers perform better than classifiers using traditional ML methods in most cases, but not always. Second, with dense word embeddings, classifiers perform better than classifiers using one-hot text features. Third, incorporating other numerical features boosts classification performance.

This paper considers methods such as classical statistical analysis and traditional ML techniques that, prior studies have shown, have relatively better performance, such as Logistic Regression, SVM, NB, and RF, and uses them as the baseline. On the side of DL methods, this

study adopts simple methods like MLP, classical sequence model like RNN, and the most recent Transformer method, taking Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT) as exemplary algorithms for RNN and the Transformers, respectively.

To make the findings more general, the author uses four datasets from the hospitality industry. The first two datasets are adopted from previous studies: one is pseudo, following Ott et al. (2011, 2013); the other is a real-life dataset from Yelp, following Mukherjee et al. (2013). The last two are newly scraped from Yelp. Also, among the four datasets, two of them are class balanced, the other two are imbalanced. In terms of features, the current study collects major public available features from Yelp but does not collect all detailed features which are not easily accessed by an outsider, such as IP address, detailed demographics, and review history.

The structure of this paper is as follows: section 2 is the literature review, section 3 describes the data, section 4 documents and discusses the methodology and results, and section 5 concludes.

2 Literature Review

Two streams of literature form the base of this study: review spam motivation and impact, and fake review detection approaches. The first stream shows the presence and impact of online review spam, which establishes the need for a system to help viewers to differentiate truthful reviews from fake. The second stream offers various classification methods to develop an automatic fake review detection system with high performance, from traditional statistical analysis and ML methods to the most updated DL techniques.

Recently, with introducing word embeddings and breakthroughs in sequence models, DL

approaches have been widely used to learn semantic representations for NLP tasks, achieving highly competitive results. A few studies have shown that fake review detection classifiers incorporating DL techniques can get over 90% accuracy (Jain et al., 2019; Ren & Ji, 2017; Shahariar et al., 2019; Zhao et al., 2018). It is easy to see the potential advantages of DL for spam detection. First, DL algorithms use multiple hidden layers for automatic feature extraction and combination, capturing complex semantic information that is difficult to get when using the traditional feature extraction approach. Second, DL algorithms usually take dense word embeddings as inputs, easing the problem arising from sparse features. But research on comparing the fake review classification performance between classifiers using DL and using ML are hardly seen. Table 1 shows a quick overview of literature related to a comparative study on fake review detection.

Table 1 Comparative study (DL vs ML) on fake review detection

Studies	Topic	DL methods used	ML methods used	Datasets used	BERT use	Beyond Text Features use
Barsever et al. (2020)	Lie detection	BERT+BiLSTM	N.A.	Ott et al. (2011, 2013)	Yes	No
Kennedy et al. (2019)	Contextualized opinion spam detection	MLP, CNN, LSTM, BERT	SVM	Ott et al. (2011, 2013) Balanced Yelp Dataset (Rayana and Akoglu, 2015)	Yes (only on Ott et al. dataset)	Yes
Kumar, J. (2020)	Fake review detection	N.A.	SVM, RF	Balanced Yelp_Liu Dataset (Mukherjee et al. (2013))	No	Yes
Shahariar et al. (2019)	Fake review detection	MLP, CNN, LSTM	SVM, NB, KNN	Ott et al. (2011, 2013)	No	No
This study	Fake review detection	MLP, LSTM, BERT	SVM, NB, RF	Ott et al. (2011, 2013) Yelp_Liu Dataset (Mukherjee et al., 2013) One Newly Mined Yelp Dataset Balanced newly mined Yelp Dataset	Yes (only on newly mined large Yelp dataset)	Yes

To the author's knowledge, one closest work to this study is the research by Shahariar et al. (2019). Their study shows that DL techniques can get higher accuracy with more data but less feature engineering than ML algorithms. In their study, DL models include MLP, CNN,

and LSTM, and ML models include SVM, NB, and K-Nearest Neighbor (KNN). The difference between their study and this work is four-fold: first, other than only using text-based and reviewer-based features, this study also includes other numerical features like product-based and psycholinguistic attributes and almost all combinations of these features; second, in contrast to only experiments on two small already available datasets with hundreds or thousands of samples, this study also experiments on two newly mined large datasets with hundreds of thousands of samples; third, to have an idea about how the most recent and powerful BERT model can apply to fake review identification tasks, this study tries and reports the performance of a fine-tuned BERT model on one dataset; fourth, this study reports the comparison outcome for two imbalanced datasets. Their study and Kennedy et al. (2019) only report experimental comparison results for balanced datasets.

2.1 Fake Review, Incentives, and Impacts

Fake reviews are not new to the world. Studies from both academia and industry acknowledge fake reviews exist as a downside of e-commerce circumstances. Despite the efforts of review platforms to filter suspicious reviews, the share of fake reviews is about 15-30% (Lappas et al., 2016; Luca & Zervas, 2016). Per a local consumer review survey done in 2020 (*Local Consumer Review Survey*, 2020), four out of five U.S. consumers believe they've read a fake review in 2019, with thirty-three percent saying they'd spotted multiple. The BBC claims that since the Covid-19 lockdown in March 2020, there has been a thirty percent rise in the proportion of suspicious reviews on Amazon between March and August according to an analysis site, ReviewMeta ("Black Friday on Amazon," 2020). Trustpilot, a global consumer review website, claims that online reviews play a critical role in internet trust, and fake reviews caused the average American consumer to waste an average of $125 in 2019 while shopping (*Five-Star*

Fraud, 2020). Research (Mayzlin et al., 2014) points out that the consumers fooled by fake reviews may make suboptimal choices, and that untruthful reviews may lead consumers to lose trust in truthful reviews.

Recent literature in economics and marketing has examined the problem of fake reviews. Research (Mayzlin et al., 2014) examines promotional reviews involving the hotel industry through differences between platforms: one is Expedia.com which only allows verified customers to post; the other is TripAdvisor.com on which anyone can post. Several articles use evidence of fake reviews for a single platform, either using filtered reviews on Yelp (Luca & Zervas, 2016), reviews with no record of purchase for a private label retailer (Anderson & Simester, 2014), or records of purchased reviews on Amazon from Facebook groups of fake review buyers(He et al., 2020).

Luca and Zervas investigate the economic incentives for a restaurant business to commit review fraud on Yelp (Luca & Zervas, 2016). Their study finds roughly sixteen percent of restaurant reviews on Yelp are suspicious: a restaurant is more likely to commit review fraud to itself and its competitors when its reputation is weak or it faces increasing competition, and when the cost of committing fraud is not high. Mayzlin et al. point out independent hotels with single-unit owners get the highest gain, but branded chain hotels with multiunit owners earn the lowest gain from promotional reviewing, partly because of the different levels of the potential cost of committing review fraud (Mayzlin et al., 2014).

Lappas et al. investigate the vulnerability of individual businesses to fake review attacks by using data from millions of hotel reviews across seventeen cities on TripAdvisor, showing that, in certain markets, just fifty fake reviews are sufficient for an attacker to surpass any of its competitors in terms of visibility(Lappas et al., 2016).

While no perfect mechanism to eliminate all review fraud is available now, several primary mechanisms can help reduce review fraud. The first and the most direct approach is to develop fake review detection algorithms further, allowing review platforms, review readers, and marketers to identify the truthfulness of the reviews with high accuracy. The second approach is to enhance the cost of committing review fraud. For instance, in May 2020, an e-commerce website was required to pay $350,000 to settle United States Federal Trade Commission (FTC) charges alleging deceptive rankings and fake reviews (*FTC Finalizes Settlement in LendEDU Case Related to Deceptive Rankings and Fake Reviews*, 2020). As an alert, introducing business spinning whose fraud activity is caught will also help review readers to be more vigilant to the reviews for this business, as Yelp is doing. Third, a straightforward and efficient way to reduce the prevalence of fake reviews is only allowing verified customers to leave a review (Mayzlin et al., 2014).

This study focuses on the development of the first approach: exploring fake review detection classifiers and learning the impact of adopting DL algorithms.

2.2 Fake Review Detection Methods

Extensive literature in computer science has also examined fake reviews, focusing on identifying ways to detect fake reviews (J. Kumar, 2020; N. Kumar et al., 2018, 2019; Rayana & Akoglu, 2015; Wu et al., 2017), as well as evaluating the effectiveness of fake review attacks (Lappas et al., 2016).

Fake review detection is essentially a binary classification problem, that is, to develop an algorithm that can tell if a review is fake or non-fake, taking a group of independent variables (predictors) as inputs. Various methods apply to this area. Only a small portion of studies work on the unsupervised or semi-supervised learning algorithms using unlabeled data (Li et al., 2014;

Mukherjee et al., 2013; Rout et al., 2017). Most research explores and develops supervised learning algorithms on training labeled data, using various approaches from basic to state of the art.

The simplest approach to detect fake reviews should be Logistic Regression, one of the most basic and classical statistical analysis algorithms to solve a classification problem. The most common spam review detection classifiers are based on supervised ML algorithms, which require labeled datasets to train and then classify the class to which each review belongs. The supervised ML algorithms this study uses include NB, RF, and SVM. After text analysis adopted DL techniques a few years ago, a few neural network-like deep structured algorithms were used to detect fake reviews, such as MLP, LSTM, and BERT.

This subsection discusses several classification algorithms to build up a theoretical background for this current study.

2.2.1 Classical Statistical Analysis Classifier: Logistic Regression

The Logistic Regression can be understood simply as finding β parameters that best fit:

$$y = \begin{cases} 1 & when \quad \beta X + \epsilon > 0 \\ 0 & o.w. \end{cases}$$

where:

y represents the class a review belongs to. 1 means the review is a fake one; 0 means non-fake. $\{y_1, y, \dots, y_i, \dots\}$; y is observable.

X represents a group of predictors related to one specific review plus one constant item; $\{x_0 = 1, x_1, x_2, \dots, x_j, \dots\}$; X is also observable.

β represents a group of correspondent coefficient parameters to X; the true value of β is unobservable but can be estimated as $\hat{\beta}$ by training on a sample dataset.

ϵ is an error item distributed by the standard logistic distribution, whose cumulative distribution function is $\frac{1}{1+e^{-\epsilon}}$; similar to β, the true value of ϵ is unobservable but can be estimated by a training dataset.

Based on the logistic distribution ϵ follows, the probability of a review is fake or non-fake can be calculated:

$$P\ (y = 1|\ X;\ \beta) = \frac{e^{-\beta X}}{1 + e^{-\beta X}}$$

$$P\ (y = 0|\ X;\ \beta) = \frac{1}{1 + e^{-\beta X}}$$

Then a threshold is set to distinguish fake or non-fake class, such as 0.5. When the probability of a review is fake is above the threshold, it will be estimated as a fake review, otherwise predicted as non-fake. Since the true value of β is not known, estimated coefficients $\hat{\beta}$ should be used with known predictors X to get the estimated probability of being a fake review, $\hat{y}$. After that, the class of a review can be estimated by comparing $\hat{y}$ with the threshold. With the assumption that every instance is independent of each other, the Logistic Regression algorithm usually takes the maximum likelihood estimation (MLE) method or minimizing a log loss function, cross-entropy, to obtain the coefficient estimators, $\hat{\beta}$. MLE and cross-entropy are stated as follows:

MLE:

The Likelihood of the training dataset is a joint likelihood of each instance which is assumed independently Bernoulli distributed to each other:

$$L = \prod_i l_i = \prod_i P\ (y_i = 1|\ X_i;\ \beta)^{y_i} * \left(1 - P\ (y_i = 1|\ X_i;\ \beta)\right)^{1-y_i}$$

Considering the log function is monotonically increasing, the log-likelihood replaces

likelihood without loss or bias of information in the estimating process to lower down the computational burden by replacing quadrature by summation. So, the log-likelihood is as follows:

$$\log L = \log\left(\prod_i P(y_i = 1|X_i;\beta)^{y_i} * \left(1 - P(y_i = 1|X_i;\beta)\right)^{1-y_i}\right)$$

$$= \sum_i y_i \log\left(P(y_i = 1|X_i;\beta)\right) + (1 - y_i)\log\left(1 - P(y_i = 1|X_i;\beta)\right)$$

$$= \sum_i y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

Minimizing the Loss Function:

The Cross-entropy of the distribution y relative to a distribution $\hat{y}$ over a training dataset is defined as follows:

$$H = -\sum_i y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

Since $\log L = -H$, when $\hat{\beta}$ maximizes log-likelihood, it also minimizes cross-entropy.

Once $\widehat{\beta}$ is obtained, the specific Logistic Regression classifier can be used to predict which category a new sample belongs to, by comparing the threshold and the calculated probability of a review being fake or non-fake.

2.2.2 Traditional Machine Learning Classifiers

To solve a binary classification problem, various machine learning algorithms can apply. This study focuses on three algorithms: NB, RF, and SVM, based on the proposed classification methods in previous research.

2.2.2.1 Naïve Bayes (NB)

Naïve Bayes (NB) is a probabilistic classification algorithm based on the Bayes theorem. It

applies in both binary and multiclass contexts.

$$P(y = c \mid X) = \frac{P(y = c) * P(X|y = c)}{P(X)}$$

Where c denotes the class, {non-fake, fake}, or use a pair of numbers such as $\{0, 1\}$ for a binary task. The denominator is effectively constant since the values of X are provided. And the numerator equals the joint probability $P(y = 1,\ X)$.

With the assumption that features are independent to each other, the above formula can be simplified as follows, by using the chain rule for repeated applications of conditional probability:

$$P(y = c \mid X) \propto P(y = c, X)$$

$$\propto P(y = c) * \prod_j P(x_j | y = c)$$

where $\propto$ means the left side is proportional to the right side.

And, respectively,

$$P(y = 1| X) \propto P(y = 1) * \prod_j P(x_j | y = 1)$$

$$P(y = 0| X) \propto P(y = 0) * \prod_j P(x_j | y = 0) .$$

To construct a classifier, NB algorithm needs to work with a decision rule, one of which is commonly used is to pick the most probable hypothesis. That is to say, the classifier should maximize posteriori (MAP). The corresponding classifier, a Bayes classifier, is the function that assigns a class label such as fake (when $y = 1$) or non-fake (when $y = 0$) as follows:

$$\hat{y} = \underset{(c=0,1)}{\operatorname{argmax}} \left(P(y = c) * \prod_j P(x_j | y = c) \right)$$

The first item of the equation, the class prior, $P(y = 1)$ or $P(y = 0)$, can be estimated by calculating an estimate for the class probability from the training set. For the whole

training set, the NB classifier aims to maximize the joint posteriori and assign fake or non-fake to each instance, that is

$$\underset{(c=0,1)}{\operatorname{argmax}} \prod_i \left(P(y_i = c) * \prod_j P(x_j | \, y_i = c \,) \right)$$

To estimate the parameters for distribution of X, an assumption of distribution should be made. For continuous features, it usually assumes x_j is distributed according to a normal distribution; for discrete features, multinomial and Bernoulli distribution are widely assumed.

Once the parameters are estimated, the specific NB classifier can be used to predict which category a new sample belongs to, by using the parameters' estimations and the values of features, X. That is the category in which the sample can have a larger posterior, in the context of fake or non-fake classification.

2.2.2.2 Random Forest (RF)

RF is another widely used binary classification technique. The foundation of the RF algorithm is the Decision Tree method (DT). A tree is built by splitting the entire data set, constituting the root node of the tree, into subsets—which constitute the successor children. In other words, DT classification depends on providing a hierarchical decomposition of the training data space, in which the tree nodes are labeled by features, X, and the branches between them are labeled by the weight that represents the occurrence of feature in the training data, and finally, the leaves are labeled by class names. The division of the data space is done recursively until the leaf nodes are reached. For nodes of numerical features, data is partitioned by the comparison outcome between data feature value and the set value, that is greater than, less than, or equals to. For nodes of categorical features, data is partitioned by its category or presence or absence. The order of the nodes is assigned according to the

importance score of the data features while designing the tree. The decision tree splits the nodes over available attributes and then selects the split which results in the most homogeneous sub-nodes. The splitting is completed when the subset at a node has all the same values of the target variable like fake or non-fake, or when further splitting adds no value to the predictions.

RF is a collection of DTs, using the same method of constructing decision trees. Multiple trees are constructed independently and parallel. A random subset of the set of features is approached by every node of the tree while training on an independent tree. Only one randomly chosen subset of the entire set of features is accessible to each node of the tree (Liaw & Wiener, 2002). All the training instances examined with substitution are utilized while constructing the forest of decision trees.

To predict the category to which a new instance belongs, just use the known predictors and follow the structure of the feature forest, and then find the category.

2.2.2.3 Support Vector Machine (SVM)

SVM is one of the most robust and popular binary classification algorithms Elmurngi and Gherbi, *An Empirical Study on Detecting Fake Reviews Using Machine Learning Techniques*., aiming to assign each instance to only one class out of two, like fake or non-fake. An SVM maps training samples to points in space to maximize the width of the gap between the two categories. The same space is used to map new instances and predict which class the instance belongs to, based on which side it falls.

For instance, a linear SVM classifier separate samples that fall in a k-dimensional space (k is the dimension of X) with a (k-1)-dimensional hyperplane. Among many hyperplanes, SVM chooses the best one that represents the largest separation between two classes. The chosen

hyperplane makes the distance from it to the nearest sample on each class maximized.

Different from Logistic Regression, to conveniently derive the algorithm behind, SVM uses $\{-1,\ +1\}$ instead of $\{0,\ 1\}$ as the class values. For a linear SVM classifier, it needs to meet the condition that

$$f(X_i) = \beta X_i = \begin{cases} \geq +1, & if\ y_i = +1 \\ \leq -1, & if\ y_i = -1 \end{cases}$$

This can be rewritten as $y_i * \beta X_i \geq +1, \quad for\ each\ instance\ in\ the\ training\ set.$

Define X_i as X_+ when $\beta X_i \geq +1$, and X_- when $\beta X_i \leq -1$, and choose the marginal instances with $\beta X_+ = +1$and $\beta X_- = -1$ as the fake and non-fake support vectors, then the margin is given by

$$\frac{\beta}{\|\beta\|}(X_+ - X_-) = \frac{1}{\|\beta\|}(\beta X_+ - \beta X_-) = \frac{2}{\|\beta\|}.$$

Thus, learning the SVM can be formulated as

$$\underset{\beta}{\operatorname{argmax}} \frac{2}{\|\beta\|} \triangleq \underset{\beta}{\operatorname{argmin}} \|\beta\|^2$$

$$s.t. \quad y_i * \beta X_i \geq +1$$

Additionally, to accommodate misclassified instance, a slack parameter $\xi_i \geq 0$ is introduced. When

$$\frac{\xi_i}{\|\beta\|} > \frac{1}{\|\beta\|},$$

the ith instance is misclassified. Furthermore, to show how much one wants to avoid misclassifying each instance, a regularization parameter, $C > 0$, is introduced to the optimization function:

$$\underset{\beta, \xi_i}{\operatorname{argmin}} \|\beta\|^2 + C \sum_i \xi_i$$

$$s.t. \quad y_i * \beta X_i \geq +1 - \xi_i \quad \triangleq \quad \xi_i = \max(0, 1 - y_i * \beta X_i)$$

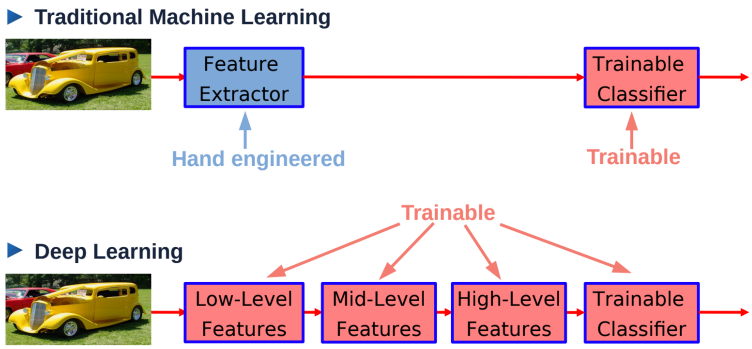
where smaller C usually gets larger margin at the cost of misclassification, and the opposite situation happens to larger C, which leads to fewer misclassification at the cost of smaller margin. After the C is set, β can be estimated and the hyperplane will be found.

For a new instance, the category to which it belongs can be predicted, by using the estimated β and the values of features, X, or comparing the location of the data point with the hyperplane.

2.2.3 Deep Learning Classifiers

Broadly speaking, deep learning (DL) is part of machine learning (ML) methods. One characteristic that makes DL methods distinguished from other traditional ML methods is based on artificial neural networks with representation learning. From Figure 1, LeCun (2020) vividly shows the mentioned difference.

Figure 1 Traditional Machine Learning vs Deep Learning



Source: Yann LeCun (2020) (*001-Intro.Pdf*)

Although the idea of neuron networks to build intellectual machines has been through ups and downs since its birth in the 1940s, it did not get widely used, especially in the commercial field, until the 2010s even with several breakthroughs such as the development of backpropagation (BP), the cheap, multi-processor Graphics cards or Graphics Processing Units (GPUs) and others. By outperforming alternative traditional ML methods in numerous

important applications, DL has finally attracted wide-spread attention and industrial applications since then (Y. LeCun, 2019; Yann LeCun et al., 2015; Schmidhuber, 2015).

Yann LeCun et al. claims, "A deep-learning architecture is a multilayer stack of simple modules, all (or most) of which are subject to learning, and many of which compute non-linear input–output mappings" (Yann LeCun et al., 2015). In other words, DL methods are representation-learning methods that extract features both simple and low as well as high and abstract, using multiple hidden layers of neural networks. For classification tasks, higher layers of representation amplify aspects of the input that are important for discrimination and suppress irrelevant variations (Yann LeCun et al., 2015).

In terms of model structure, there are three kinds of layers in a DL model:

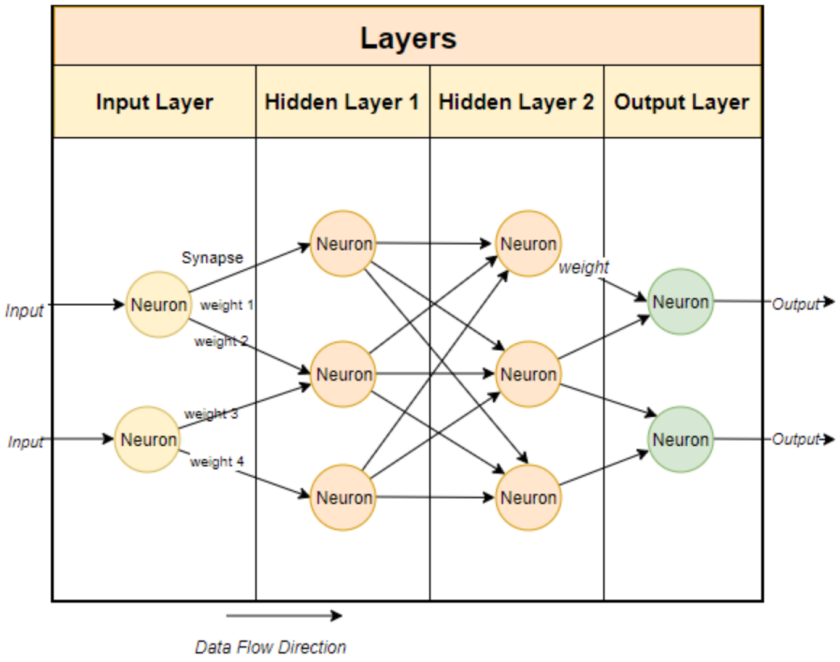
First is the input layer. It feeds the inputs to the model and thus is at the very beginning of the workflow for an artificial neural network. Second is the output layer. It produces the outputs and exports results out of the model, and thus is the outlet of the workflow for the network. Third are hidden layers located between the input and output layer, forming the body of the network. In a DL model, there needs to be at least two hidden layers for the model to be called 'deep'. The hidden layers perform nonlinear transformations of the inputs, and the outcome of the last hidden layer is taken as the input of the output layer.

Each layer consists of a collection of neurons. Each neuron contains information about input, weight, and bias, and a nonlinear activation function that is applied to produce the result of the neuron by feeding the linear combination of input and bias. Then, the result of the neuron will be part of the input for neurons of the next layer. In this way, the neuron uses a nonlinear activation function and is connected to neurons of the neighboring layers via links. Each link has a weight, which determines the strength of the influence that one neuron has on

other related neurons.

The framework for a DL model aiming to detect fake reviews has three steps. First, the input layer carries neurons of initial features, which are collected from various sources such as review content, reviewer-related, and product-related data. Second, the output layer exports results via neurons about the probability that one review is fake or non-fake. Third, the hidden layers run multiple nonlinear transformations, develop the most important and abstract representation of the inputs, and then export through neurons to the output layer. With different architecture used across the layers, different DL models are built up. Figure 2 shows a diagram of a DL model, composed of one input layer with two neurons, two hidden layers with three neurons in each, and one output layer with two neurons. Neurons of different layers are connected while neurons of the same layer are not connected.

Figure 2 A Diagram of a DL model



Source: Malik, "Understanding Neural Network Neurons."

In around 2010, a type of convolutional neural network (CNN) architecture was proved to

be able to produce vector representations of words and yield record-breaking results on various natural language processing tasks. The NLP community was initially skeptical about whether to adopt the CNN method and lacked confidence in the benefits of adoption so that the CNN method did not become dominant in the NLP area until 2016 (Y. LeCun, 2019). Various DL architectures have been created since then, such as the MLP, CNN, RNN, embeddings from language model (ELMo), generative pre-training (GPT), and many more. In October 2018, a new language representation model called BERT was created, and Google AI researchers show that the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks, such as question answering and language inference (Devlin et al., 2019). Upon its release, BERT was believed to be one of the most important breakthroughs in the NLP area.

Because they include extracting features from review content, fake review detection tasks fall in the NLP area. Meanwhile, fake review detection is also beyond NLP because detection usually engages in many other attributes such as reviewer behavior and product information.

DL techniques tend to solve the classification problem end to end. Text review classification has benefited from the recent resurgence of DL architectures, which potentially lead to high accuracy with less feature engineering. For spam review detection, DL algorithms require much more training data than traditional ML algorithms. But with the help of new methods such as Word2Vec that obtain better vector representations for words, DL algorithms can improve the accuracy of classifiers, even with the same amount of data that ML algorithms use. To the author's knowledge, applying DL methods to fake review detection tasks can be studied further, because only a few papers have focused on them (Jain et al., 2019; Kennedy et al., 2019; Ren & Ji, 2017; Ruan et al., 2020; Shahariar et al., 2019; Shukla

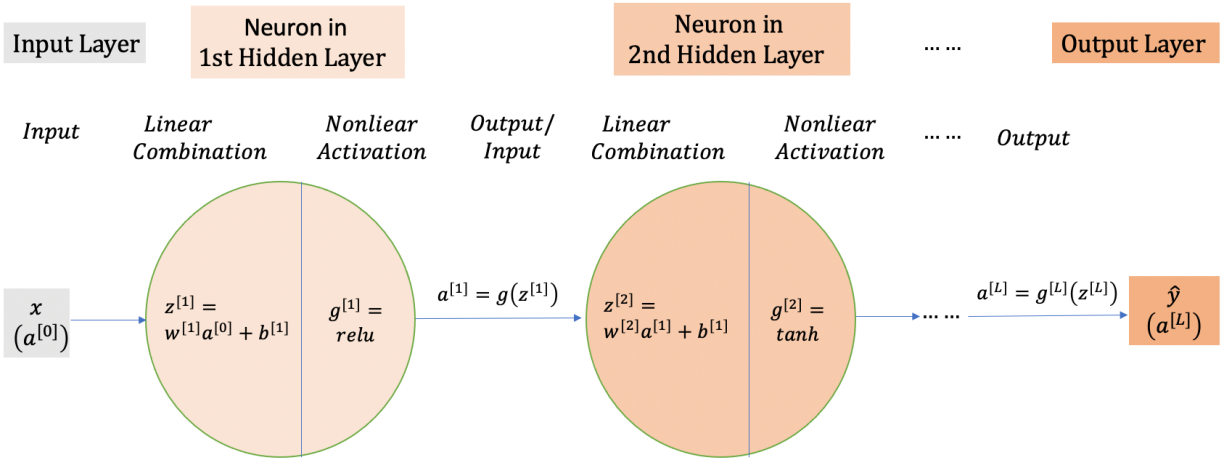
et al., 2019; Tang et al., 2016).

This subsection covers three DL architectures, MLP, LSTM, and BERT, which range from simple to complex.

2.2.3.1 Multilayer Perceptron (MLP)

In this study, MLP models refer to the 'vanilla' DL models with multiple hidden layers of the perceptron. The number of neurons in each hidden layer and the number of layers are set by researchers. Below is a diagram of an MLP model. Each neuron of the hidden layers consists of two parts: linear combination and nonlinear activation.

Figure 3 A Diagram of an MLP model



As Figure 3 shows, $z^{[l]}, w^{[l]}, a^{[l-1]}, b^{[l]}, a^{[l]}$ *and* $g^{[l]}$ denote respectively the linear combination of result, weight, input, bias, output and the activation function of the lth hidden layer. $a^{[l-1]}$ and $a^{[l]}$ are respectively the outputs of the $l-1th$ and the lth layers. At the beginning of the flow, it's input x, which can also be put as $a^{[0]}$, and in the end, $a^{[L]}$ is the output of the last hidden layer, the Lth layer, and can also be represented as $\hat{y}$.

The most common activation function g_s used in classification tasks includes sigmoid, tanh, rectified linear unit (Relu) (Nair & Hinton, 2010), and Gaussian error linear unit (Gelu)

(Hendrycks & Gimpel, 2020). The sigmoid function is the most widely used activation function of the last hidden layer. The learning occurs by changing weights after each batch of data is processed, based on the amount of error in the output compared to the expected result. This process is carried out through backpropagation (BP). BP repeatedly adjusts the weights to minimize the loss between the output of the DL model ($\hat{y}$) and the true value (y). The way to adjust weight is called gradient descent (GD), which is to change each weight in proportion to the first-order derivative of the error to that weight, then the minimum of the error term can be found, given the non-linear activation functions are differentiable. Details about BP and GD can be found in the paper authored by Rumelhart, Hinton & Williams (1986).

In this study, the author used multiple MLP models with 2-5 hidden dense layers, 1-128 neurons for each layer, in different experimental contexts.

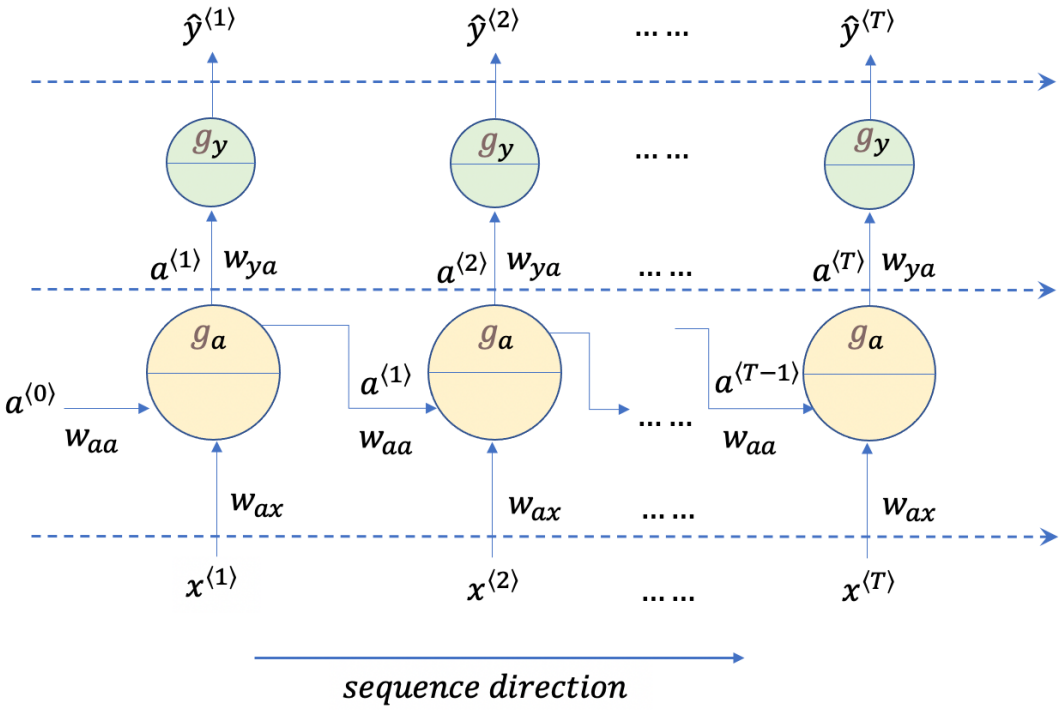
2.2.3.2 Long Short-Term Memory (LSTM)

LSTM was introduced in 1997 (Hochreiter & Schmidhuber, 1997). It is one type of sequence model, a specialized Recurrent Neural Network (RNN), which is introduced to handle sequential inputs like sentences or paragraphs of text. RNN is especially suitable for some NLP tasks like language translation, text sentimental analysis, and text classification because the input of each instance in those cases is often a series of words, sentences, or paragraphs. For a T-word-length sequential input $\left(x^{\langle 1\rangle}, \dots, x^{\langle t\rangle}, \dots, x^{\langle T\rangle}\right)$, the hidden state for each input words is $\left(a^{\langle 1\rangle}, \dots, a^{\langle t\rangle}, \dots, a^{\langle T\rangle}\right)$, the output is $\left(y^{\langle 1\rangle}, \dots, y^{\langle t\rangle}, \dots, y^{\langle T\rangle}\right)$, the following equations and Figure 4 express the relations between these three:

$$a^{\langle t\rangle} = g_a\left(w_{aa}a^{\langle t-1\rangle} + w_{ax}x^{\langle t\rangle} + b_a\right)$$

$$y^{\langle t\rangle} = g_y\left(w_{ya}a^{\langle t\rangle} + b_y\right)$$

Figure 4 A Diagram of an RNN model



where g_a is the activation function of the hidden state, g_y is the function of the output layer; w_{aa} is the hidden-to-hidden weights, w_{ax} is the input-to-hidden weights, w_{ya} is the hidden-to-output weights; b_a and b_y are bias items for hidden layer or output layer respectively.

Similarly, BP and GD are used to find the minimum of the error item of an RNN model.

One major downside of the RNN model is its short-term memory because of the vanishing gradient problem and thus relevant information from earlier may be left out. LSTM functions just like RNN, but it is capable of learning long-term dependencies using mechanisms called gates: input gate, forget gate, and output gate. These gates are different tensor operations that can learn which information should add to or remove from the hidden state. Also, an LSTM unit maintains a memory cell state. The cell state transfers and carries relative information, from the latest or much earlier time steps only if it's relevant, all the way down the sequence chain, and discards irrelevant information along the way. Information gets added or removed

to the cell state via gates through the sequence. The gates are different neural networks that decide which information is relevant to keep or forget during training. Further details about LSTM can be found in the paper authored by Hochreiter & Schmidhuber (1997).

When applying the LSTM model to the fake review detection task, several studies show it has the potential to outperform existing models. Ruan et al. propose a manual fake review detection model which is ensembled by LSTM, SVM, and AdaBoost methods, by combining the information of the reviewer's account and geolocation (Ruan et al., 2020). Tang et al. base on an LSTM model and introduce one more gating function to alleviate information loss in case of a very long sequence context (Tang et al., 2016). Shukla et al. use a bidirectional Gated Recurrent Unit (GRU) model, which is similar to LSTM, and demonstrate a significant improvement in accuracy in contrast to traditional logistic regression and random forest and human evaluators (Shukla et al., 2019).

The LSTM algorithm used in this study is an extended version, which is called Bidirectional LSTM. To further improve model performance o sequence classification tasks, Bidirectional LSTM trains two instead of one LSTM on the input sequence, in which one is from the beginning to the end and the other is in the opposite order. The author tried 1-2 Bidirectional LSTM layers, 128 neurons for each layer, in different experimental contexts.

2.2.3.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT was developed by Google in late 2018 and is said as conceptually simple and empirically powerful (Devlin et al., 2019). By obtaining new state-of-the-art results for eleven major NLP tasks, BERT proves its amazing power and shows that it can be fine-tuned to perform almost all major NLP tasks. Based on the publication, the keys of BERT are summarized as follows:

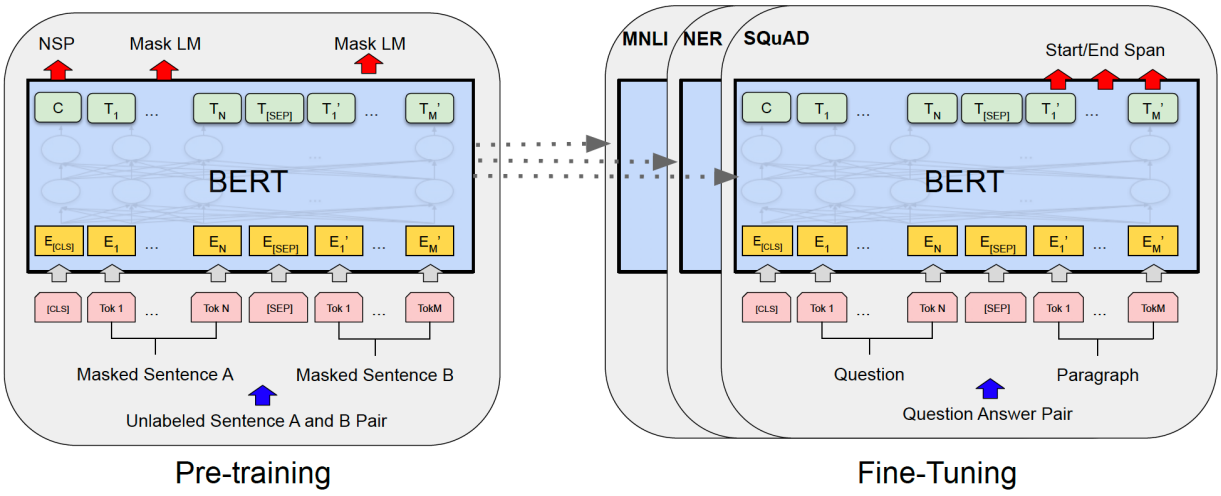
Two steps are involved in the BERT framework: pre-training and fine-tuning.

In the pre-training step, taking a "masked language model" (MLM) pre-training objective enables BERT to capture relative information from the left and the right context, which allows the authors to pre-train a deep bidirectional Transformer. Also, a "next sentence prediction" task (which can be seen as a binary classification NLP task) enables BERT to jointly pre-trains text-pair representations. The corpus used to pretrain the model is enormous: BooksCorpus (800M words) plus English Wikipedia (2,500M words). In terms of BERT's model architecture, it is a multi-layer bidirectional Transformer encoder based on the original implementation described in the famous paper entitled "Attention is all you need" (Vaswani et al., 2017). Two pre-trained models are reported: one base and the other large. Even for the base one, the number of total parameters is as large as 110M. The resulting library consists of 30,000 token vocabularies, with two special classification tokens, ([CLS]) and ([SEP]).

In the step of fine-tuning, the BERT model is first initialized with the pre-trained parameters, that is, pre-trained parameters are imported into the model. And then the model is trained on the labeled sampling data, at last, all the parameters are finished with fine-tuning. When applying the BERT model, we simply plug in the task-specific inputs and outputs into BERT and finetune all the parameters end-to-end. Compared to pre-training, fine-tuning is relatively inexpensive.

Figure 5 shows the pre-training and fine-tuning procedures when using BERT.

Figure 5 Overall pre-training and fine-tuning procedures for BERT



Source: Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018).

As of today, only a very few studies apply the BERT model to the fake review detection task. A few studies report new state-of-the-art classifiers using fine-tuning BERT (Barsever et al., 2020; Kennedy et al., 2019), achieving over ninety percent accuracy on a widely used crowd-sourcing review dataset, which is introduced by Ott et al. (2011, 2013). Jawahar et al. conduct an in-depth error analysis of RoBERTa, which is a BERT-based classifier to detect text generated by text generative models (TGM). Their study shows that the classifier can be attacked hard by using simple schemes such as replacing characters with homoglyphs and misspelling some words. These two attacks can reduce the detector's recall from 97.44% to 0.26% and to 22.68%, respectively (Jawahar et al., 2020). In this study, the result of using a fine-tuned BERT model to detect fake reviews for a large field dataset is reported.

2.3 Feature Extraction

Extracting and constructing features from data including review text is one key step when developing a fake review detection classifier. Features can be categorized into three main groups. The first group is composed of review features. Word counts, term frequencies, and

word embeddings are widely used review features. The second group refers to reviewer features. The fundamental reasoning behind this is that the spammers who manipulate review system may have behavior characteristics which are different from normal review writers. Features can be extracted from the reviewer's information such as his personality, viewing behavior such as the number of reviews and average review length, and his social network. The third group is product features, like the literal description of the product, features of the product, and so on.

Most prior studies only use review features in building fake review detectors, and some recent studies also use reviewer features. It shows by a few papers that using a combination of features from different groups to train the classifier can perform better (Jindal & Liu, 2008; Mukherjee et al., 2013; Sun et al., 2016). Intending to construct and compare various detectors which integrate an algorithm with features, this study includes hotel information in addition to review features and reviewer features.

This subsection discusses all three groups of features.

2.3.1 Review-based Features

Review features are extracted directly from the review content, like word counts and word frequencies, psycholinguistic attributes based on word libraries, and word embeddings. Review-based features are broader, however, and include several other numerical features like review rating, the number of useful/funny/cool votes for a review, photos used in a review, and others.

One challenge for researchers in the NLP area was how to convert a corpus, which is a collection of words, sentences, paragraphs, and essays, into text features and use them as inputs to train a model. In this subsection, the author addresses several traditional and updated

text feature extraction methods.

2.3.1.1 BOW features

One approach is Bag-of-Words (BOW), which uses word presence or word frequency of each word as a feature to represent text like review content. BOW features disregard grammar and word order information. Table 2 shows an example of BOW features where one feature relates to one word using the word's presence in the text. In application, the frequency of occurrence of each word is also widely used as a feature.

Table 2 An example of bag-of-words feature

Text	a	avoid	business	for	hotel	it	nice	not
A nice hotel for business	1	0	1	1	1	0	1	0
Avoid it, not a nice hotel	1	1	0	0	1	1	1	1

A vocabulary is comprised of all the words that show up in the corpus. For instance, in the above example, the list of words constructs a vocabulary: a, nice, avoid, hotel, business, for, it, and not. The size of the vocabulary (V) is at least as large as tens of thousands, maybe ten times more. Also, the unit, whose presence or frequency is used as a review-based feature, can be a single word, a small group of continuous characters, and a group of consecutive words, like two or three words. For instance, if we take two words as one unit in the vocabulary, the BOW features for the first review in the above table should be: ones for 'A nice', 'nice hotel', 'hotel for' and 'for business', and zeros for other units such as 'avoid it', 'it not' and 'not a', which are only covered by the second review. This type of feature is called n-gram, in which n denotes the number of words in one unit.

Another variation is called TF-IDF, which uses a numerical statistic called tf-idf to replace the presence or frequency of a word in the vocabulary. The formula to calculate TF-IDF is:

$$tf - idf = tf * idf.$$

For our fake review detection task, tf means n-gram term frequency. Usually, it means the number of times that an n-gram term occurs in a review. Presence or term frequency adjusted for document length is used to measure tf. idf measures whether the n-gram term is common or rare across the corpus, by using a log of the inverse fraction of the reviews that contain the n-gram term. So, for rarely occurring n-gram terms, the value of idf is high, and vice versa. In the BOW model, the numerical presentation of one word is a one-hot 1*V vector, consisting of all zeros for other words in the vocabulary and only one non-zero for the word itself.

All kinds of BOW features, no matter whether presence, frequency, or TF-IDF have something in common: sparseness. For each word, its one-hot vector representation using BOW methods would be […0, 1, 0,…] alike, taking word presence as an example. The location of the word in the vocabulary results in the location of 1 in the vector, and all the other locations in the vector are zeros. Since there is only one non-zero in the vector, people call it a one-hot vector. In this study, the author call text features using BOW methods as one-hot text features. For a review, the number of unique words, n, is small by contrast to the large size of the vocabulary, V. So, the representation of a review is a 1*V matrix, and the matrix is very sparse. The matrix consists of a lot of zeros for words that are absent in the review and only n non-zeros for those present words. Furthermore, these word features only focus on the occurrence of the word, ignoring the positional and semantic information.

This study uses both word counts and TF-IDF as one-hot text features.

2.3.1.2 Word Embeddings

To overcome the limitation of BOW features, modern DL-based NLP utilizes one hidden layer to train and optimize the vectorization and preserve positional information for each word in

the vocabulary. The representation of a word is no longer a scalar (0/1 or a specific number) but a dense vector with a much lower dimension, 1*N (N << V). The first word-embedding model utilizing neural networks was created by a Google research team (Mikolov et al., 2013). Since then, word embeddings features appear in almost every NLP model used in practice today due to their effectiveness. The model using word embeddings can catch the semantic information in the text and make it possible to perform mathematical operations.

Two ways exist for obtaining word embeddings: either use extant pre-trained word embeddings like the pre-trained one from the BERT model; or build your word embeddings from scratch, using the specific text data under study. For the BERT word embeddings, the dimension of a word representation is 768, trained and extracted from an enormous corpus of 3,300 million words through training around 110 million parameters. For self-made word embeddings, the size of the word embedding dimension is a hyperparameter to set. Dimensions ranging from fifty to three hundred are widely used.

Both pre-training BERT word embeddings and the ones trained from scratch are used in this study. The author sets the dimensionality of the latter embeddings as 128. This study uses word embeddings with DL algorithms only.

2.3.1.3 Psycholinguistic Features

Linguistic Inquiry and Word Count (LIWC) software (Pennebaker et al., 2015), a popular automated text analysis tool, is a popular way to extract psycholinguistic features from text. LIWE has been used to analyze deception since its emergence, and studies have shown its combination with other features makes detectors perform better (Ott et al., 2011, 2012).

In particular, the second and latest version, the LIWC2015 counts and groups nearly 6,400 words into 93 default library categories. Each category is composed of words with similar

psychological or linguistic meanings (Pennebaker et al., 2015). The default LIWC2015 can be categorized into four groups. First, the summary and detailed language variables: overall aspects of the text (e.g., analytical thinking, clout, authenticity, emotional tone, and percentage of words in the text that are pronouns, articles, auxiliary verbs, etc.). Second, psychological processes: Includes all social, emotional, cognitive, perceptual, and biological processes, as well as anything related to time or space. Third, personal concerns: features relate to work, home, leisure, etc. Fourth, informal language and punctuation categories: primarily filler and agreement words, periods, commas, etc.

2.3.1.4 Other review-derived features

Besides the above text features, researchers also examine other review-based features such as review length, time, rating, and so on (Jindal & Liu, 2008; Li et al., 2014; Mukherjee et al., 2013). Their findings show that these features are beneficial to detect fake reviews.

When using the review-related features, this study integrates fifteen features as inputs such as review length, review rating, review polarity, review subjectivity, word length, readability, and others.

2.3.2 Reviewer-based Features

Many studies have shown that using only review-based features, fake review detectors perform less excellent. Reviewer-based features have been proved to have the potential to help improve the detector's performance (Jindal & Liu, 2008; J. Kumar, 2020; Li et al., 2014; Mukherjee et al., 2013).

Reviewer behavior features include profile features and behavior features. The number of written reviews, reviewer id, joining date, social network connection with others, and other features in profile. Behavior features usually refer to review rating distribution, the number of

times of being the first review writer, the review content similarity, the maximum number of reviews in a day, and so on.

In this study includes reviewer-based features like friends count, the number of reviews posting, and the number of photos sharing.

2.3.3 Other Features

It has been shown that including product-based features is helpful to enhance fake review detection performance. Sun et al. propose a convolutional neural network model to integrate the product-related review features through a product word composition model (Sun et al., 2016).

This study includes a few hotel-based features such as hotel review count, average hotel review lengths for each rating level.

2.4 Classification Performance Evaluation

Performance evaluation metrics are fundamental in assessing the quality of learning methods and learned models (Ferri et al., 2009), and form a base in directing the training process and achieving better classifiers.

For a binary classification task, the classification outcome by a given classifier can be outlined by a confusion matrix as Table 3:

Table 3 Confusion Matrix of a Fake Review Classification Model

		Predicted Class	
		Non-fake	Fake
True Class	Non-fake	TN	FF
	Fake	FN	TF

where:

TF (true fake) and TN (true non-fake) correspond to the number of true fake and true non-fake reviews. That is, TF is the number of reviews that were correctly classified as fake, and TN is the number of reviews that were correctly classified as non-fake. On the other hand, FF (false fake) and FN (false non-fake) correspond to the number of false fake and false non-fake reviews, which were incorrectly classified to the other class other than the correct one.

And the performance metrics used in this study are based on these four numbers.

2.4.1 Classification Accuracy

The first metric is classification accuracy, defined as

$$Accuracy = \frac{TN+TF}{TN+TF+FN+FF},$$

measuring the proportion of being correctly classified instances among the total number of instances examined. It is the most widely used diagnostic tool, especially in a class-balanced context, to evaluate a single classifier and compare with different classifiers. But it could be misleading when the metric of accuracy is used for a highly imbalanced classification task, because even no skill classifier can achieve a high accuracy due to the imbalance of the data by only predicting the majority class. The lowest and highest value of $Accuracy$ is 0 and 1, respectively. A classifier with higher accuracy is evaluated as a better classifier.

2.4.2 F-score, Precision, and Recall

The second metric is the F-score, which is the harmonic mean of precision and recall. So before diving into F-score, let us look at precision and recall, which are defined as below:

$$Precision = \frac{TF}{TF + FF}$$

$$Recall = \frac{TF}{TF + FN}$$

where precision measures the proportion of reviews that are predicted as fake that belongs to the fake class, and recall summarizes how well the fake class is predicted, that is the percentage of true fake reviews are correctly predicted as fake. Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. Usually, precision and recall scores are not discussed in isolation. Evaluation description is always put like the precision level of 0.80 at a recall level of 0.75.

Or combine precision and recall into one measure, F-score, which is originally defined as

$$F_{score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{TF}{TF + \frac{1}{2}(FF+FN)},$$

and used as another metric of accuracy. Like $Accuracy$, F_{score}has its boundary values of 0 and 1. Unlike $Accuracy$, F_{-score} does not take TN, true non-fake, into account, and only focuses on the fake class. Also, it gives the same weight to precision and recall, but for many real-world classification tasks, the costs of different types of misclassifications are different. So, a positive real factor β is introduced such that recall is considered β times as important as precision. The formula will be changed accordingly. Since the original F_{-score} values precision and recall with the same weights, F_{-score} is usually called as $F1_{score}$.

2.4.3 Macro-F1score

To accommodate precision and recall of both fake and non-fake class, the macro-F1 score is introduced to use, especially for a production application. There exist two different formulas to calculate macro-F1score. The one used in this study and widely used in academic research and production circumstance (Ren & Ji, 2017), the arithmetic mean of class-wise F1-scores, is significantly more robust towards the error type distribution (Opitz & Burst, 2019). Per the above formula of F1-score, two F1-scores, one for fake and the other for non-fake, can be calculated. The one for fake class, F1-score-fake, is identical to the above F1-score formula,

and the other for the non-fake class is

$$F1_{nonfake} = \frac{2}{\frac{1}{Precision\_nonfake} + \frac{1}{Recall\_nonfake}} = \frac{TN}{TN + \frac{1}{2}(FF+FN)},$$

by using TNs to replace the TFs in the prior formula. And then average these two, macro-F1 score, the third metric used in this study, is then defined as:

$$macro - F1_{score} = \frac{1}{2}\left(F1_{fake} + F1_{nonfake}\right).$$

The macro-F1 score is often used in situations where classes are unevenly distributed, like the fake review detection task (Opitz & Burst, 2019), since it covers both majority and minority classes. Same as $F1$, $macro - F1_{score}$ has boundary values between 0 and 1. The classifier with a higher $macro - F1_{score}$ is considered better.

2.4.4 AUC

The fourth metric used in this study is the area under the curve (AUC). Overall, it is regarded as a good metric used in many applications when good class separation is pursued, including spam filtering, fraud detection, and others (Ferri et al., 2009). A curve in common use is Receiver Operating Characteristic (ROC), which can evaluate the ability of binary classifiers to distinguish classes.

A ROC curve is a diagnostic plot to summarize the behavior of a model by calculating the false positive rate (usually used as horizontal axis) and true positive rate (usually used as vertical axis) for a set of predictions by the model under different thresholds. It shows the ability of a probabilistic classifier to rank the positive instances relative to the negative instances. For the task under study, the true positive rate is the true fake rate, which is defined as,

$$TrueFakeRate = \frac{TF}{(TF+FN)},$$

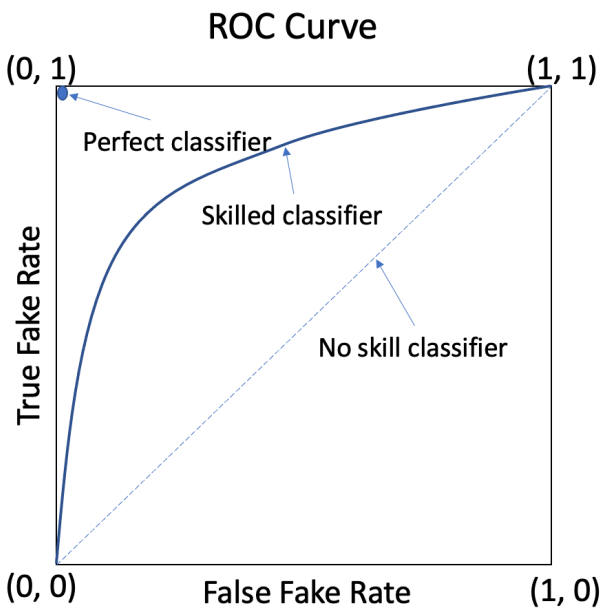
and the false positive rate is false fake rate, which is defined as:

$$FalseFakeRate = \frac{FF}{(FF+TN)}.$$

The threshold is set by researchers from 0.0 to 1.0. The predicted class of each instance is decided by comparing the predicted probability with every given threshold. Then the above four numbers, TF, TN, FF, and FN are obtained, and TrueFakeRate and FalseFakeRate can be calculated as per the above formulas. A point can be drawn in the plot. As the threshold moves, the point moves accordingly. Finally, the points are connected to form a curve. In the context of finite instances, it is a step function, which approaches a true curve as the number of instances approaches infinity. Figure 6 shows a ROC curve.

A classifier that has no skill (e.g. predicts the majority class under all thresholds) will be represented by a diagonal line from the bottom left to the top right. Any points below this line show the classifiers that are even worse than no skill classifier. A perfect model will be a point in the top left of the plot when TrueFalseRate is 1 and TrueFakeRate is 0. In this way, the ROC Curve is helpful to diagnose a model.

Figure 6 ROC Curve



The area under the ROC curve (ROC AUC) provides a single score to summarize the plot that can be used to compare models. A no-skill classifier has a score of 0.5, whereas a perfect classifier has a score of 1.0. ROC AUC is considered a good evaluation not only for one single classifier but also for comparing performance between classifiers. The higher is ROC AUC, the better the classifier is.

When applying to imbalanced contexts, ROC AUC is arguably less informative. Cortes and Mohri show that ‘‘the average AUC is monotonically increasing as a function of the classification accuracy, but that the standard deviation for uneven distributions and higher error rates is noticeable. Thus, algorithms designed to minimize the error rate may not lead to the best possible AUC values” (Cortes & Mohri, 2003).

3 Data

One big obstacle for fake review detection research is the lack of a ground-truth dataset. This study examines four review datasets involving the hotel industry, in which the first two are

widely used in the field of fake review identification, and the third dataset is newly scraped from Yelp.com. The fourth dataset is derived from the third one by balancing class distribution.

3.1 Ott dataset

The author directly borrows the dataset from Ott (*Deceptive Opinion Spam Corpus Myle Ott*, n.d.; Ott et al., 2011, 2013). Ott et al. gathered and introduced the dataset in 2011 (positive reviews) and 2013 (negative reviews). After that, Ott et al.'s dataset quickly became widely used in identifying fake reviews. The dataset consists of two parts. The first 800 reviews with truthful tags come from review platforms including TripAdvisor, Expedia, Orbitz, Hotels.com, Priceline and Yelp. There are 20 positive and 20 negative truthful reviews for each of the 20 popular hotels in the Chicago area, resulting in 400 positive reviews and 400 negative reviews in total. The second 800 reviews with a deceptive tag are from Amazon Mechanical Turk (MTurk) workers, following the offered golden standard. Except for the origin, the structure of the second 800 reviews is the same as the first part.

The unique feature of this dataset is that the fake reviews are created as per a pre-defined gold standard, and thus the reviews labeled as deceptive are untruthful without doubt.

But the Ott dataset does have a major shortcoming, which has been shown in studies by Mukherjee et al., (2013). The latter study shows that although fake, the MTurk generated reviews are not real-life fake reviews on a commercial website. Firstly, the psychological state of mind of Turkers when writing shows the difference from that of real fake review writers who have real businesses to promote or to demote. Secondly, the word distributions of truthful and deceptive reviews are very different, while the real-life spammers did a good job by using words with almost equal frequency as words used by real reviewers. This indicates that the Ott

dataset is not representative of the real-life fake reviews, though it is easier to split into fake and non-fake.

Therefore, it is necessary for this study to not only use the Ott dataset but also include datasets from real-life review platforms like Yelp.com. Furthermore, the Ott dataset does not include information about reviewers or hotels, so, the identification algorithm trained from it may be less general. All other three datasets used in this study come from Yelp.com.

3.2 Yelp datasets

In actual life, it is unlikely to know the ground truth about the fakeness of an online review, because researchers cannot directly observe if a review is fake. Furthermore, previous work shows that human identify fake reviews almost by chance (Jindal & Liu, 2008; Ott et al., 2011, 2013). So, it is difficult to obtain a labeled dataset with absolute certainty for a supervising learning algorithm to detect fake reviews. Thanks to Yelp, researchers have proxy data to work on.

Yelp is one popular review platform, on which people post reviews about various businesses such as hotels, restaurants, and many other services. Even during the pandemic, as of Q3 2020 (*Yelp - Company - Fast Facts*, 2021), Yelp self-reported that on average more than 32 million app unique devices connect to it every month, and over 220 million new reviews are available. In contrast, that Google accepts star-rating-only reviews, Yelp insists on requiring review text besides star rating, which adds value to reviews. One recent study from a researcher of FTC (Raval, 2020), finds that on average, Yelp reviews contain 593 characters, which is more than double Google's 250 characters.

What makes Yelp review data become the closest labeled data, when the ground-truth data is not available, originates from a unique Yelp feature: its review filtering system,

Recommendation Software. The automatic filtering system differentiates which review Yelp recommends and which not. The system was put in place soon after Yelp's big launch in 2005. Though the filtering algorithm is not public information, the results of the filtering algorithm are. Only the reviews that pass the filtering system and are considered worthy of being recommended will get published on Yelp's main listings. Those reviews which are filtered and regarded not recommendable from the perspective of Yelp, are not listed literally on the main business page but can be seen through a link on the page. The filtered reviews do not count towards calculating a business's average star-rating. Yelp (2009) disclosed the filtering system's purpose remains the same: to protect consumers and business owners from fake, shill or malicious reviews (Inc & Monday, 2009). Also, Yelp admits that though the system has evolved over the years, it is not perfect. Legitimate review content may sometimes get filtered and illegitimate review content may also get published. Yelp introduces that reviews belonging to three main types are considered not recommended and will get filtered: fake ones originating from the same computer, biased ones that may be written by a friend, and real ones but posted by a less established user(*Why Would a Review Not Be Recommended? | Support Center | Yelp*, n.d.). As of September 30, 2020, among all reviews posted by Yelp users, 70% were published, 22% not recommended and 8% removed for violating the site's Content Guidelines (*Yelp - Company - Fast Facts*, 2021).

Luca and Zervas (2016) validate using Yelp's algorithmic filtered reviews as a proxy for review fraud, showing the cheating businesses have much higher rates of algorithmically identified fake reviews relative to the authors' main sample. To put terminology consistently, this study uses fake and non-fake rather than filtered and non-filtered, although the latter one is more correct than the former.

In this current study, three datasets from Yelp are discussed below.

3.2.1 Yelp_Liu dataset

The first Yelp dataset used in this study was part of the dataset which was originally mined by Mukherjee et al. (2013) when the authors researched what Yelp's review filter might be doing. The latest data is from September 2012. The author got access to it with permission from Dr. Bing Liu, one author of the above paper, and only used hotel reviews from the original set of datasets, which also included restaurant reviews. I will call it as Yelp_Liu dataset.

This Yelp_Liu dataset used in this study includes 779 fake and 5078 non-fake reviews with text across 85 hotels in the Chicago area, posted by 5132 reviewers. Besides review text, the dataset includes the numbers of useful/cool/funny votes which a review gets. Table 4 shows the dataset is class imbalanced, with around 13.3% of filtered reviews. Within the same star-rating reviews, relatively more fake reviews have a top or worst star rating: for all 1-star and 5-star reviews, 31.15% and 17.40%, respectively, are fake. For the middle star rating, like 3-star and 4-star, fake reviews are less common, accounting for, on average, less than 8% of total reviews of these star level ratings.

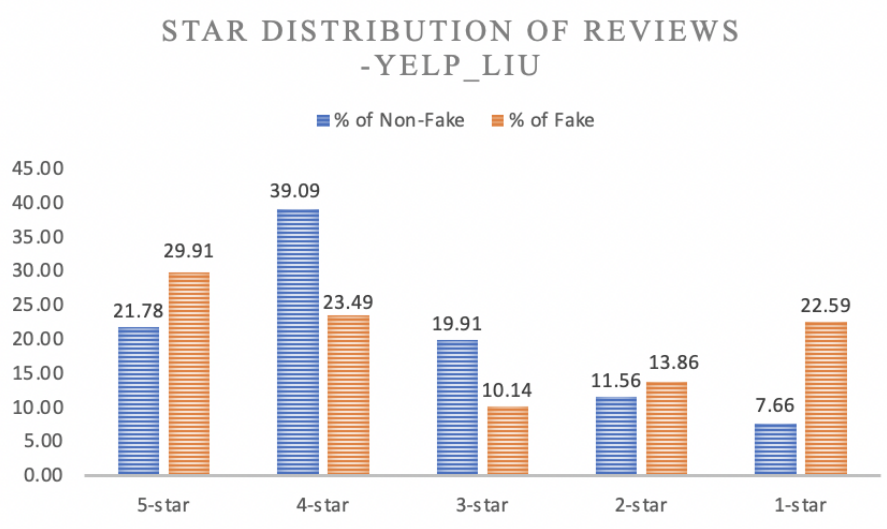
Table 4 Review Rating Distribution of Yelp_Liu Data

Review Rating	No. of Reviews	% of Reviews	No. of Non-Fake	% of Non-Fake	No. of Fake	% of Fake	% of Fake in all
5-star	1339	22.86	1106	21.78	233	29.91	17.40
4-star	2168	37.02	1985	39.09	183	23.49	8.44
3-star	1090	18.61	1011	19.91	79	10.14	7.25
2-star	695	11.87	587	11.56	108	13.86	15.54
1-star	565	9.65	389	7.66	176	22.59	31.15
Sum	5857	100	5078	100.00	779	100.00	13.30

Also, within one class, the star distribution of reviews is quite different, which is shown in both Table 4 and the histogram graph, Figure 7. More than half of fake reviews have

extremely high and low ratings, in which 29.91% is the top 5-star while 22.59% gets the worst 1-star, while only less than one-third of non-fake reviews get the best or worst star ratings, 21.78% and 7.66%, respectively.

Figure 7 Review Rating Distribution of Yelp_Liu Data



3.2.2 Newly Mined Yelp dataset

The second dataset from Yelp was manually mined. A total of 606,850 hotel reviews are from Yelp. The dataset consists of 3,792 hotels across 16 cities, with dates ranging from October 2004 to September 2020. To collect both fake and non-fake labeled reviews from Yelp, a program was developed accordingly. The data collection program includes two tasks. It firstly navigates Yelp's home page and collects all the hotels (3,792 hotels) in a list of pre-defined cities, i.e., Las Vegas, New York, Los Angeles, and 13 other cities. Secondly, the program iterates the hotels and collects all of their fake and non-fake reviews. Some reviewer-based and hotel-based features are also collected for further use in the study. After removing non-English reviews, the sample size is decreased to 598,658.

The resulting dataset is class imbalanced, and I will call it as the Yelp_all dataset. Table 5 shows 10.21% of reviews are labeled as fake. Within the same star-rating reviews, relatively

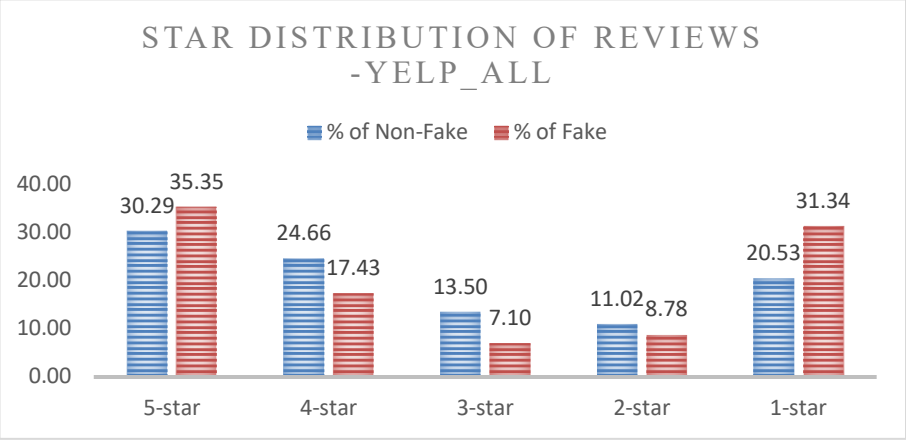
more fake reviews have a top or lowest star rating: 14.79% of 1-star reviews and 11.71% of 5-star reviews are fake. For the three middle star ratings, fake reviews account for less than 6%-9% in each.

Table 5 Review Rating Distribution of Yelp_all Data

Review Rating	No. of Reviews	% of Reviews	Non-Fake No.	% of Non-Fake	Fake No.	% of Fake	% of Fake in all
5-star	184422	30.81	162823	30.29	21599	35.35	11.71
4-star	143205	23.92	132556	24.66	10649	17.43	7.44
3-star	76915	12.85	72578	13.50	4337	7.10	5.64
2-star	64599	10.79	59237	11.02	5362	8.78	8.30
1-star	129517	21.63	110367	20.53	19150	31.34	14.79
Sum	598658	100.00	537561	100.00	61097	100.00	10.21

Similar to the review rating distribution of the Yelp_Liu dataset, the distribution of the Yelp_all dataset in fake or non-fake class is quite different, which is shown in both Table 5 and Figure 8. Almost two out three fake reviews have extremely high and low ratings, in which 35.35% is best 5-star while 31.34% gets the worst 1-star, while only one-half of non-fake reviews get the best or worst star ratings.

Figure 8 Review Rating Distribution of Yelp_All Data



In contrast to the Yelp_Liu dataset, a new trend can be seen from Yelp_all dataset. It

seems on average, more and more Yelp reviews, no matter if they are fake or non-fake, tend to express extreme experience, which can be seen from the rapidly growing proportion of 1- and 5-star reviews.

The third Yelp dataset is balanced and formed from the Yelp_all dataset, by randomly selecting non-fake to match the fake reviews, to construct a balanced dataset. After non-English reviews are removed, the sample size of this balanced Yelp dataset is 136,717, of which 68,378 are non-fake and 68,339 are fake.

In all, this study trains multiple classifiers on the above four datasets, among which the Ott dataset is small, balanced, and pseudo, Yelp_Liu is small, imbalanced, Yelp_all is large and imbalanced, and Yelp_balanced is not small and balanced. The latter three datasets, which come from Yelp, are all real-life.

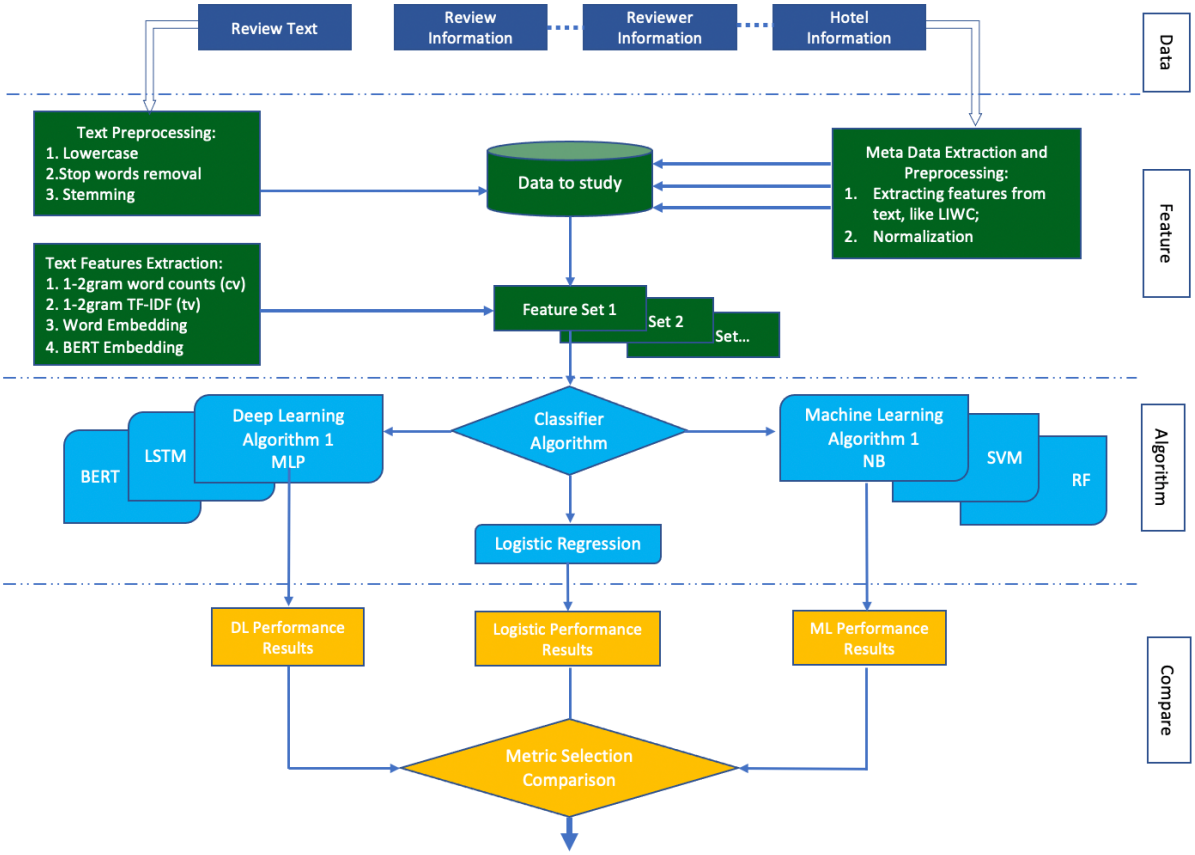
4 Methodology and Results

This section discusses the methodology and the results of this study.

4.1 Methodology

To compare the performance of fake review classifiers, this study employs different classification algorithms including DL, ML, and Logistic Regression, and trains classifiers on data with various features. Figure 9 shows the study's workflow, which includes four phases.

Figure 9 Workflow and Phases of this study



The first phase is all about data. The four datasets used in this study are explained in section 3. All the information is grouped into four categories: review text content, review-based, reviewer-based, and hotel-based. The second phase is about features: preprocessing and extraction. For the unstructured text data, standard NLP preprocessing measures are taken

such as stop words removal, lowercase English letters conversion, and words stemming. After the text corpus is cleaned, text features are extracted in ways discussed in section 2.3.1 and section 2.2.3.3 (for BERT): n-gram word counts, term frequencies, word embeddings, and BERT embeddings. Additionally, LIWC2015 is used to extract psycholinguistic metrics from the text, and several other numerical features are extracted by using predefined functions in Python. For other structured numerical metadata about the reviewer or hotel, it will be normalized with other numerical review-based features after the data is split into training, validating, and testing parts. When all the features are ready, a combination of different features forms a feature set. For instance, psycholinguistic features from LIWC2015 belong to review-based features, they can combine with word count features that belong to text features, and further combine with the reviewer-based and hotel-based features. So can word embeddings. The feature sets used can be seen in the tables about the findings of each dataset.

The third phase is about the algorithm: DL methods, ML methods, and the traditional Logistic Regression method. All algorithms are discussed in section 2.2, including DL methods like MLP, LSTM, and BERT, ML algorithms like SVM, NB, and RF, and Logistic Regression. After a feature set is given and hyperparameters are set, a classifier using a specific algorithm will be trained on the training portion of sample data and then validated on the validation portion. Thus, all parameters are estimated to form a model with known estimated values of parameters. Then, the author maps the model to the testing data, compares the predicted class with the true label, and gets the confusion matrix, computes, and obtains the value of performance metrics, which are discussed in section 2.2.4. Lastly, the fourth phase is about performance comparison between classifiers, which is determined by using appropriate performance evaluation metrics. The result of the comparison discloses the main

findings of the article.

The next subsection presents the details about the above process.

4.2 Codes and running platform

All the processing is done on either local Jupyter Notebook or Google Collaboratory Notebook (Colab). When the local hardware resource is good enough to process training on small datasets like the Ott dataset and the Yelp_Liu dataset, and part of small model training on the Yelp_balanced dataset, local Jupyter Notebook is used. When higher computational demand is required to run model on large datasets like the Yelp_all dataset, and part of large model training on the Yelp_balanced dataset, Colab is used.

The author uses Python version 3.7 when coding. For classifiers with algorithms of Logistic Regression, SVM, NB, and RF, the author wrote codes by using embedded standard packages, like NumPy, pandas, scikit-learn, CSV, and so on, and the installed valid packages, like Matplotlib, Keras, Tensorflow and so on. For classifiers using DL algorithms, the author uses the most famous frameworks for DL, Keras, which is a DL application programming interface (API) written in Python or R, running on top of the machine learning platform TensorFlow. Sequential class and Functional class were used in different cases: Sequential class is viable for single-input, single-output stacks of layers; Functional class is viable for two or more input situations, for instance, when word embeddings and meta-features are both fed into the model.

4.2.1 Hyperparameters about features

For word counts and TF-IDF features, hyperparameters about features include uni-bigram method, vocabulary size of 20,000, and max-df of 0.9.

For word embeddings extracted from the local corpus, hyperparameters about features

include vocabulary size of 20,000, sequence-length of 256, and embedding dimensions of 128 and 64 are used respectively.

For pre-tuned BERT word embeddings, no extra arguments about features need to specify.

4.2.2 Hyperparameters about model structures

The author uses packages with hyperparameters either by default or by pre-defined values when necessary.

For Logistic Regression and three ML algorithms, the author uses all hyperparameters or arguments by default when methods are given. For SVM, the linear_SVC method is used, for NB, multinomial and gaussian distribution are tried. The splitting percentage between training and testing is 80:20, and 20% of training is assigned as validating.

For DL algorithms, the author did not make models very complicated and deep because the aim is about finding the best classifier. So the models do not have many hidden layers nor very complicated structure, instead, only incorporate multiple layers when running the chosen algorithms, MLP, LSTM, and BERT.

For MLP, the number of hidden dense layers is set between 2 to 5, the size of each hidden layer is set between 1 to 128, and dropout methods are also tried but no all MLP models remain dropout layers. Figure 10 shows an example for an MLP classifier used in the study: in total 5 hidden dense layers are there; 128/50/25/10/1 are the number of neurons of each layer; in total, more than 2.5 million parameters are trained.

Figure 10 A Diagram of Model Structure of an MLP model

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_5 (Dense)              (None, 128)               2572032
_________________________________________________________________
dense_6 (Dense)              (None, 50)                6450
_________________________________________________________________
dense_7 (Dense)              (None, 25)                1275
_________________________________________________________________
dense_8 (Dense)              (None, 10)                260
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 11
=================================================================
Total params: 2,580,028
Trainable params: 2,580,028
Non-trainable params: 0
```

For LSTM method, a stack of two LSTM layers and only one LSTM layer are tried to form the first part of inputs, nlp_input, then a flatten or a dense layer was tried after LSTM layer(s), then meta-features are fed to the model as the second part of inputs, and lastly a few dense hidden layers were employed. Figure 11 is an example for an LSTM classifier used in this study. First layer is about word embeddings, then only one layer of LSTM layer word embeddings layer, then dropout layer, then meta_input is fed, and a joined input layer is formed after combining nlp_input with meta_input, and lastly three consecutive dense hidden layers to export the outcome.

Figure 11 A Diagram of Model Structure of an LSTM model

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
nlp_input (InputLayer)          [(None, 256)]        0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 256, 128)     2560000     nlp_input[0][0]
__________________________________________________________________________________________________
bidirectional (Bidirectional)   (None, 256)          263168      embedding[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 256)          0           bidirectional[0][0]
__________________________________________________________________________________________________
meta_input (InputLayer)         [(None, 112)]        0
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 368)          0           dropout[0][0]
                                                                 meta_input[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 64)           23616       concatenate[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 32)           2080        dense[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 16)           528         dense_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1)            17          dense_2[0][0]
==================================================================================================
Total params: 2,849,409
Trainable params: 2,849,409
Non-trainable params: 0
```

For both MLP and LSTM, the activation function used in all dense layers in MLP and LSTM models is Relu, and the one used in the output layer is softmax or sigmoid. The learning rate is used by default, 0.001, and binary cross-entropy and Adam were used as the loss function and optimizer, respectively. The batch size are assigned to be 8, 128, 256 according to the sample size, and the number of epochs is at least 10 and up to 256, decided by meeting the rule of non-overfitting. To avoid over fitting, early stopping techniques are used.

For BERT, first it demands huge computation resources. That's why the author runs the model on Colab, using more powerful GPU that offered by Google. But the GPU on Colab is also limited. It shows that even with Colab's professional subscription, the author frequently got warning that says the procedure was crushed because of lack of available resource. So, the author only experiment on the Yelp_all dataset, and randomly selected 10% to 20% samples to fine-tune the BERT model. The base BERT pre-tuned model has around 110 million parameters. At the stage of fine-tuning, the author tries a series of hyperparameters, such as

128 or 256 as the maximum sequence-length, 4 as the number of epochs, 8, 16, 32, 64 as the numbers of batch size, Relu and Gelu are tried to be the activation function of the dense hidden layer, and binary cross-entropy and Adam are used as the loss function and optimizer, respectively. Figure 12 is an example for a BERT classifier used in this study. First, get the pre-trained BERT model as first part of the input layer and train it on the sampling data, then combine additional features, and last get the outcome after a dropout layer.

Figure 12 A Diagram of Model Structure of a BERT model

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_word_ids (InputLayer)     [(None, 240)]        0
__________________________________________________________________________________________________
input_mask (InputLayer)         [(None, 240)]        0
__________________________________________________________________________________________________
segment_ids (InputLayer)        [(None, 240)]        0
__________________________________________________________________________________________________
keras_layer (KerasLayer)        [(None, 768), (None, 177853441   input_word_ids[0][0]
                                                                 input_mask[0][0]
                                                                 segment_ids[0][0]
__________________________________________________________________________________________________
additional_feature (InputLayer) [(None, 7)]          0
__________________________________________________________________________________________________
pooled_with_additional (Concate (None, 775)          0           keras_layer[0][0]
                                                                 additional_feature[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 775)          0           pooled_with_additional[0][0]
__________________________________________________________________________________________________
output (Dense)                  (None, 2)            1552        dropout[0][0]
==================================================================================================
Total params: 177,854,993
Trainable params: 177,854,992
Non-trainable params: 1
__________________________________________________________________________________________________
```

4.2.3 Performance evaluation metrics selection

As discussed in section 2.2.4, comparing classification performance needs appropriate evaluation performance. In this study, for the balanced datasets, the Ott dataset, and the Yelp_balanced, all four metrics, accuracy, F1score, macro-F1score, and AUC are applicable. For the imbalanced datasets, as discussed in the previous section, accuracy should not be used as a major evaluation metric, and ROC-AUC is arguably less informative in an imbalanced context, so F1score and the macro-F1 score are more applicable to compare.

4.3 Results

To each dataset, the author applies at least one DL algorithm, besides Logistic Regression and ML algorithms. The main results from each dataset are discussed in the following subsections. An overall discussion section follows.

4.3.1 Results of the Ott dataset

The author tries three ways to extract text features: word counts, TF-IDF, and word embeddings from raw review content text or preprocessed review content. To lower down the dimension of the word vector, the author also gets truncated text features by using a dimension reduction method: singular value decomposition (SVD). Then, the author trains multiple classifiers using different algorithms and feeding text features alone or the combination of text features and LIWC features. To get an idea about how word embeddings work in a classifier, the author also replaces the above-mentioned text features, word counts and TF-IDF, with word embeddings and trains.

Table 6 shows the main results on the Ott dataset. The first four rows show that LIWC features are informative for detection tasks, the middle rows indexing from 5 to 9 show the best results for each algorithm, and the last row shows the results of using the MLP algorithm with word embeddings and LIWC features.

Table 6 Main Results of the Ott Dataset

Index	Algorithm	Feature Group				Evaluation Metrics			
		Word Count	TF-IDF	LIWC	Word Embeddings	Macro_F1	F1	Accuracy	AUC
1	SVM	-	-	×	-	82.79%	82.20%	82.81%	82.78%
2	RF	-	-	×	-	82.72%	81.48%	82.81%	82.71%
3	Logistic	-	-	×	-	82.48%	81.82%	82.50%	82.46%
4	MLP	-	-	-	×	83.75%	83.85%	83.75%	90.67%
5	SVM*	-	×	-	-	92.49%	92.26%	**92.50%**	92.50%

6	RF	-	×	-	-	89.38%	89.38%	89.38%	89.41%
7	NB*	-	×	-	-	92.18%	**92.40%**	92.19%	92.27%
8	Logistic	-	×	-	-	92.49%	92.21%	92.50%	92.46%
9	MLP*	-	×	-	-	**92.50%**	92.31%	**92.50%**	92.49%
10	MLP*	-	-	×	×	91.87%	91.61%	91.87%	**97.57%**

*means the classifier gets the best performance, at least on one metric: SVM/NB/MLP+(TF-IDF).

Training on the same data, Ott et al. (2011) show that psycholinguistic features from LIWC2007 are helpful to detect fake reviews, and its combination with bigram can slightly improve the classifier's accuracy from 89.6% to 89.8%. In contrast, the best classifiers in this study show accuracy as high as 92.5%, with an increase of 2.7%.

In terms of the research questions about algorithm and the way of extracting text features, the author finds from reading the experimental outcome from a small balanced pseudo dataset like the Ott dataset:

1. LIWC features alone are beneficial for building a fake review classifier, but when combined with other features it may or may not help.

As seen from the first three rows, psycholinguistic features from LIWC2015 alone are helpful to detect fake reviews with accuracy above 82.50%. But when it comes to features combination, this study finds in some cases, the incorporation of LIWC features may or may not improve the classifier's performance. For instance, feeding TF-IDF features only instead of a combination with LIWC features leads to the best results of accuracy across algorithms including MLP, various DL algorithms, and logistic regression.

2. Word Embeddings alone are a bit more informative than LIWC features but they seem to be less informative than TF-IDF features.

Feeding word embeddings only as inputs, the MLP classifier obtains an accuracy of

83.75%, which is better than feeding LIWC features only, but is well behind feeding TF-IDF features.

3. Classifiers using DL and ML algorithms perform evenly.

Except for the RF classifier whose classification accuracy is below 90%, the best performance for each algorithm is around 92%. Relatively, the classifiers using SVM or MLP algorithms result in the best performance at 92.5%. It is hard to tell which algorithm, ML or DL, outperforms in the context of the Ott dataset.

4. Combination of word embeddings and LIWC features helps to boost MLP's performance.

Using the MLP algorithm and combining LIWC features with word embeddings, the classifier performs much better than using word embeddings only: the accuracy leapfrogs from 83.75% to 91.87%.

4.3.2 Results of the Yelp_Liu dataset

Based on the same Yelp_Liu dataset, Mukherjee et al. (2013) report the results of using unigram or bigram only and a combination of n-gram and Part-of-Speech (POS) / deep-Syntax: F1score is between 25.4% and 31.7%. Afterwards, studies based on the same original data usually make it balanced by randomly choosing part of non-fake reviews so a much higher F1score around 90% is reported (Kumar, 2018). Since previous research shows that the inclusion of other review-based numerical features helps to enhance performance, the author includes them as required features so that every feature set includes it.

Table 7 Main Results on the Yelp_Liu Dataset

Algorithm	Feature Group					Evaluation Metrics			
	Word Count	TF-IDF	LIWC	Word Embeddings	Review_based	Macro_F1	F1	Accuracy	AUC

MLP	×	-	-	-	×	50.21%	**29.65%**	58.70%	18.10%
Logit	×	-	-	-	×	53.33%	15.09%	84.64%	67.50%
SVM	×	-	-	-	×	**55.33%**	24.00%	77.30%	55.89%
RF	×	-	-	-	×	n.a.	n.a.	86.52%	50.00%
NB	×	-	-	-	×	52.70%	12.90%	86.18%	53.01%
MLP	-	×	-	-	×	**55.02%**	**28.09%**	71.16%	58.76%
Logit	-	×	-	-	×	51.84%	12.73%	83.62%	66.15%
SVM	-	×	-	-	×	52.01%	23.47%	72.18%	55.07%
RF	-	×	-	-	×	n.a.	n.a.	86.52%	50.00%
NB	-	×	-	-	×	52.01%	12.12%	85.15%	52.42%
MLP	×	-	×	-	×	**61.27%**	**36.62%**	76.96%	71.50%
Logit	×	-	×	-	×	54.02%	18.32%	81.74%	68.76%
SVM	×	-	×	-	×	54.25%	19.12%	81.23%	53.89%
RF	×	-	×	-	×	n.a.	n.a.	86.52%	50.00%
NB	×	-	×	-	×	52.01%	12.12%	85.15%	52.42%
MLP	-	×	×	-	×	**57.90%**	**31.67%**	74.23%	71.47%
Logit	-	×	×	-	×	55.40%	22.37%	79.86%	71.02%
SVM	-	×	×	-	×	54.25%	19.12%	81.23%	53.89%
RF	-	×	×	-	×	n.a.	n.a.	86.52%	50.00%
NB	-	×	×	-	×	52.70%	12.90%	86.18%	53.01%
MLP	-	-	×	-	×	**65.14%**	**46.04%**	75.60%	81.81%
Logit	-	-	×	-	×	60.61%	41.58%	69.80%	81.98%
SVM	-	-	×	-	×	56.17%	39.56%	62.46%	74.56%
RF	-	-	×	-	×	n.a.	n.a.	86.52%	50.00%
NB	-	-	×	-	×	54.21%	39.60%	58.87%	76.23%
LSTM	-	-	-	×	×	64.85%	40.86%	81.23%	82.13%
MLP*	-	-	-	×	×	**78.47%**	**63.06%**	89.51%	90.32%

*means the classifier gets the best performance: MLP + (Word Embeddings+Others)

In this study, the author does not manually balance the class distribution but keeps the imbalance as it is. So, the performance evaluation metrics should be macro-F1score and F1score other than accuracy. The major findings on the Yelp_Liu dataset, a small imbalanced real-life dataset, are listed in Table 7 and discussed as below:

1. Classifiers using DL algorithms perform better than other classifiers.

Across all kinds of combinations of features, DL classifiers outperform all other classifiers

using ML or logistic regression methods. For instance, the simplest DL classifiers, which use the MLP algorithm, with only two hidden layers, obtain up to 65.14% for macro-F1. In contrast, the best macro-F1 among all other classifiers using various ML algorithms is only 60.61%. When looking at F1, the same thing happens. The only exception is the classifier using the SVM algorithm: it obtains a higher macro-F1 with word counts and review-based features as the feature combination.

2. Classifiers with word embeddings outperform classifiers with one-hot encoding features.

When review-based features are pre-included, classifiers using both DL algorithms, LSTM and MLP, and word embeddings outperform all classifiers that use ML algorithms and one-hot word counts or TF-IDF. For instance, LSTM and MLP model obtains 64.85% and 78.47% for macro-F1, respectively, while all classifiers using DL algorithms only get scores under 61%. Besides, even using MLP algorithm but with one-hot features, the highest macro-F1 is also much lower, 61.27%.

3. Including LIWC features as inputs to one-hot text features improves classifiers' identification performance in most cases.

The contribution of LIWC features is very significant to MLP classifiers, less significant to Logistic classifiers, and even less to DL classifiers. For instance, when LIWC is included in word counts features, the MLP classifier's macro-F1 is boosted from 50.21% to 61.27%. A similar but much less impactful thing happens to most DL classifiers and Logsitic classifiers. The only exception happens to the SVM classifier. Things seem to be mixed; the inclusion of LIWC features could help macro-F1 increase or decrease. The macro-F1 increases from 52.01% when feeding TF-IDF only to 54.25% when feeding LIWC features together, but it

can also decrease from 55.33% when feeding word counts only to 54.25% when feeding LIWC features too.

4. Classifiers using MLP methods perform best.

The best performance is achieved when employing the MLP algorithm and training on word embedding features. The macro-F1 and F1 soared to 78.47% and 63.06% from 65.14% and 46.04%, respectively. This result again demonstrates that the combination of word embedding and LIWC assures good performance to some extent.

5. LSTM performs better than all ML methods.

The first experiment of LSTM seems acceptable. It achieves the third-best performance: macro-F1 and F1 are 64.85% and 40.86%, respectively. Although the LSTM model is left way behind the best MLP model, which uses word embedding features as inputs, it outperforms all other various classifiers which use ML algorithms.

4.3.3 Results of the Yelp_all dataset

Recall the Yelp_all dataset is imbalanced, so like for the Yelp_Liu dataset, classifiers' performance evaluation metrics should also be based on Macro-F1 and F1, avoid Accuracy, and be alert to AUC.

Table 8 Main Results of the Yelp_all Dataset

Algorithm	Feature Group							Evaluation Metrics			
	Word Count	TF-IDF	LIWC	Word Embedding	Review	Reviewer	Hotel	Macro_F1	F1	Accuracy	AUC
MLP	-	-	-	×	-	-	-	**63.84%**	**39.39%**	80.37%	81.88%
BERT	-	-	-	×	-	-	-	62.28%	27.45%	74.61%	62.28%
MLP	×	-	-	-	-	-	-	61.10%	26.86%	91.22%	93.92%
Logit	×	-	-	-	-	-	-	53.07%	27.89%	66.58%	71.10%
SVM	×	-	-	-	-	-	-	53.41%	28.07%	67.18%	65.99%
RF	×	-	-	-	-	-	-	**63.25%**	**30.93%**	91.68%	59.24%

MLP	-	×	-	-	-	-	-	51.74%	8.56%	90.37%	74.67%
Logit	-	×	-	-	-	-	-	54.27%	28.92%	68.31%	72.44%
SVM	-	×	-	-	-	-	-	53.17%	28.40%	66.28%	66.76%
RF	-	×	-	-	-	-	-	**62.94%**	**30.31%**	91.67%	59.00%
MLP	×	-	×	-	×	×	×	56.35%	17.62%	90.70%	89.72%
Logit	×	-	×	-	×	×	×	66.29%	45.02%	79.71%	89.66%
SVM	×	-	×	-	×	×	×	**67.29%**	**45.32%**	82.06%	78.85%
RF	×	-	×	-	×	×	×	63.25%	30.93%	91.68%	59.24%
MLP	-	×	×	-	×	×	×	**68.79%**	42.15%	91.54%	91.71%
Logit	-	×	×	-	×	×	×	65.70%	**45.11%**	78.07%	90.99%
SVM	-	×	×	-	×	×	×	65.21%	44.47%	77.56%	83.31%
RF	-	×	×	-	×	×	×	64.84%	33.95%	91.97%	60.30%
LSTM*	-	-	×	×	×	×	×	**76.92%**	**59.45%**	90.13%	93.46%
MLP	-	-	×	×	×	×	×	71.01%	51.25%	84.49%	89.75%

*means the classifier gets the best performance: LSTM+(LIWC+Word Embeddings+others).

Table 8 carries the findings from experimenting on the Yelp_all dataset. The author finds:

1. BERT gets good performance.

Due to limited available computational resources, the author applies the BERT embeddings to only a small portion of this imbalanced large dataset to build a BERT fine-tuned classifier. No other features are included. With multiple attempts, the author reports only one fine-tuned BERT model as follows: first, call BERT-base-uncased pre-tuned word embeddings, second, randomly select 20% of the data as the sample data, which contains 149,950 instances, then split it into training, validation, and testing parts (90: 8: 2), set max sequence length, epochs, batch size, and learning rate as 128, 4, 32, 3e-5, and use Gelu and Adam as the nonlinear activation function and optimizer, respectively.

The macro-F1 metric for the BERT model is 62.28%. On one hand, as compared to the MLP model using word embeddings trained from scratch, which obtains 63.84% on macro-F1: it is lower by 1.6%. On the other hand, in contrast to the models which only use either word counts or TF-IDF features, the BERT model performs second best, only standing behind the

RF model by less than 1% while outperforming all the others.

But, considering this BERT model trains on only 20% of the dataset, while the other models work on all sample data, it is possible for the BERT model to perform even better or outperform others by using more data.

2. RF outperforms MLP when one-hot text features only and MLP outperforms RF when other features are included with TF-IDF as inputs.

When feeding with text features only, either word counts or TF-IDF, classifiers using RF perform better than detectors using MLP. When using macro-F1 as the metric to measure, the performance difference varies from 2.15% to 11.2%.

On the occasion that includes other review-based features, performance comparing results become mixed. When the word count is used, MLP still performs poorly, but when TF-IDF is used with other features as inputs, MLP performs best when using macro-F1 to measure.

3. Including other numerical features helps classifiers perform better.

By contrast to the values of macro-F1 and F1, it shows that in most cases, classifiers with more features included perform much better than the ones with only text features as inputs. For instance, macro-F1 is boosted to 68.79% from 51.74% by including features relating to information about reviews, reviewers, and hotels.

4. Word Embeddings are more informative than one-hot text features.

For the MLP algorithm, macro-F1 of classifiers improves sharply to 63.84% from 61.10% (word counts) or 51.74% (TF-IDF), when replacing one-hot text features with word embeddings.

This finding complies with the findings on the Yelp_Liu dataset, but it seems to be opposite to the findings on the Ott dataset. Discussions are in subsection 4.3.

5. With word embeddings and other features as inputs, LSTM wins the best performance.

Obtaining macro-F1 of 76.92% and F1 of 59.45%, LSTM wins itself the best classifier, using word embeddings, LIWC features, and meta-features about reviews, reviewers, and hotels as inputs. When using the same feature set, the MLP model still performs great, surpassing 70% on both macro-F1 and F1 scores.

4.3.4 Results of the Yelp_balanced data

The fourth dataset is originated from the Yelp_all dataset, randomly selecting a small portion of non-fake reviews to make the class distribution balanced. So, all four metrics can be used safely when evaluating or comparing the detection performance.

Based on the findings discussed in subsection 4.2.3, including other features will help improve the classifier's performance. So, for the experiments on the Yelp_balanced dataset, all classifiers get trained to keep the features from reviews, reviewers, and hotels as a part of their inputs.

Table 9 Main Results of the Yelp_balanced Dataset

Algorithm	Feature Group							Algorithm			
	Word Count	TF-IDF	LIWC	Word Embedding	Review	Reviewer	Hotel	Macro _F1	F1	Accuracy	AUC
MLP	×	-	×	-	×	×	×	81.05%	81.87%	81.09%	**88.49%**
Logit	×	-	×	-	×	×	×	79.40%	80.13%	79.43%	86.98%
SVM	×	-	×	-	×	×	×	78.64%	79.12%	78.65%	78.64%
RF	×	-	×	-	×	×	×	**86.66%**	**87.20%**	**86.68%**	86.67%
MLP	-	×	×	-	×	×	×	83.49%	84.78%	83.59%	90.47%
Logit	-	×	×	-	×	×	×	83.69%	85.47%	83.89%	**90.58%**
SVM	-	×	×	-	×	×	×	83.80%	85.60%	84.00%	83.96%
RF	-	×	×	-	×	×	×	**87.23%**	**87.71%**	**87.24%**	87.23%
MLP*	-	-	×	-	×	×	×	87.58%	88.43%	87.64%	**93.99%**
Logit	-	-	×	-	×	×	×	82.98%	85.43%	83.33%	90.06%
SVM	-	-	×	-	×	×	×	81.91%	84.76%	82.36%	82.30%
RF*	-	-	×	-	×	×	×	**88.79%**	**89.30%**	**88.81%**	88.79%

MLP	-	-	-	×	-	-	-	72.87%	74.29%	72.94%	81.80%
LSTM	-	-	×	×	×	×	×	84.32%	85.94%	84.49%	**92.36%**
MLP	-	-	×	×	×	×	×	**87.65%**	**88.10%**	**87.67%**	91.44%

*means the classifier gets the best performance, at least on one metric. RF/MLP+(LIWC+others).

Table 9 carries the findings from experimenting on the Yelp_balanced dataset. The author finds:

1. The performance improvement across all classifiers is impressive and as expected.

Most of the metrics are well above 80%, and some AUCs even surpass 90%, in contrast to relatively low macro-F1 and F1 metrics scores, under 80%, for classifiers on the Yelp_all dataset.

2. Classifiers using the RF algorithm have the best performance.

Taking macro-F1 to illustrate, the highest value, 88.79%, is found in the classifier that uses LIWC features as inputs and RF as the algorithm. MLP models get up to 87.65% for the macro-F1metric when combining LIWC features and word embeddings as inputs.

3. Word embeddings work marginally better than one-hot encoding features.

When LIWC features and other meta-features remain in feature sets, the inclusion of word embeddings leads to higher macro-F1 and F1 by contrast to the score when incorporating word counts or TF-IDF as inputs. For instance, the MLP model obtains 87.65% and 88.10% for the two metrics, whereas the RF model gets 87.23% and 87.71%, respectively.

Table 10 summarizes all the experimental findings and the best classifiers of all the four datasets.

Table 10 Results of the four datasets

Dataset	**Findings on each dataset**

Ott	1. LIWC features alone are informative to build a fake review classifier, but when combined with other features it may help or not. 2. Word Embeddings alone is a bit more informative than LIWC features but less than TF-IDF features. 3. Classifiers using DL algorithms perform evenly and mixed in contrast to classifiers using ML algorithms. 4. Combining word embeddings with LIWC features helps to boost MLP's performance.
Yelp_Liu	1. Classifiers using DL algorithms perform better than other classifiers. 2. Classifiers using word embeddings outperform classifiers using one-hot encoding features. 3. Including LIWC features as inputs besides one-hot text features improves classifiers' identification performance in most cases. 4. MLP performs best. 5. LSTM performs better than all ML methods.
Yelp_all	1. BERT obtains good performance. 2. RF outperforms MLP when one-hot text features only and MLP outperforms RF when other features are included with TF-IDF as inputs. 3. Including other numerical features helps classifiers perform better. 4. Word Embeddings are more informative than one-hot text features. 5. Combining word embeddings and other features as inputs, LSTM wins the best performance.
Yelp_balanced	1. The performance improvement across all classifiers is impressive but not unexpected. 2. Classifiers using RF algorithm has the best performance. 3. Word embeddings works better than one-hot encoding features, though marginally.

4.4 Discussion

Based on the findings of each dataset, this subsection presents the answers to the research questions and discusses general findings and a few exceptions.

Table 11 Answers to the Research Questions Based on Experiment Results

Research Questions	Ott dataset	Yelp_Liu dataset	Yelp_all dataset	Yelp_balanced dataset

Deep learning (DL) outperforms machine learning (ML)?	Almost the same	Yes	Yes	No
Do word embeddings outperform one-hot text features?	No	Yes	Yes	Yes
Other numerical features are helpful to detect fake reviews?	N.A.	N.A.	Yes	Yes
Best Classifier*: Algorithm + (Features)	SVM + (TF-IDF) NB + (TF-IDF) MLP + (TF-IDF)	MLP + (Word Embeddings+ Others)	LSTM + (LIWC+ Word Embeddings+ Others).	RF + (LIWC+Others). MLP + (LIWC+Others).

*Best classifier refers to the classifier whose performance is the best, reading the evaluation metric's value.

Table 11 summarizes the answers to the three research questions for each dataset. The first research question is which algorithms perform better: DL algorithms versus ML algorithms? The fourth row of table 4.5 shows the best classifier on each dataset. By reading the algorithm the best classifiers use, this study finds that DL algorithms outperform ML models in the two actual and natural class-imbalanced datasets, the Yelp_Liu and the Yelp_all datasets. Subsection 4.2 shows in detail that the performance difference is not marginal, sometimes it is a lot. But the author also notices that DL methods underperform ML methods in the Yelp_balanced dataset, showing that the classifiers using the RF method perform best. The author's guess about the reason behind this is the limited number of hidden layers of the DL models and the shrunken number of the non-fake samples. That is the high level of the abstract features extracted may be not enough to help the DL models perform better than the RF models. Also, the author finds that DL and ML methods perform in the Ott dataset. The above reasoning may also apply to explain the even result. But the reasoning does not rule out other explanations and should be tested in future research. Furthermore, when looking closer to the classifiers with one-hot text features alone on the Yelp_all dataset, the author finds that

RF models also perform better than MLP models. From this exception, the author sees that for DL methods to showcase their potential, researchers need to avoid using one-hot text features alone but use the dense word embeddings plus other numerical features as inputs.

Also, inside DL methods, the author finds that the simple MLP model performs well, but for large actual review datasets, more delicate DL model like a LSTM has the potential to perform even better. Probably, the BERT method can improve performance more than this study reports. The main obstacle preventing BERT from using is its high demand for enormous computational resources. After overcoming it, researchers should see leapfrog progress on performance.

The second research question is which text feature extraction approach helps more for classifiers to perform better: dense word embeddings versus sparse one-hot text features? The answer is quite certain: across the three Yelp datasets and without exception, the classifiers using the dense word embeddings outperform classifiers using the sparse one-hot text features. It is clear that dense word embedding vectors are more advanced and can catch more information than one-hot text features, for embeddings not only account for the occurrence of words but also their semantic meanings. Not surprisingly, the results from the three real-life datasets showcase the above finding. The only exception happens to the Ott dataset. The reason could be that the small sample size, only 1,600 instances, is too small to construct a solid word embeddings vector for each word and thus cannot further improve the classifier's performance.

Also, this study shows that incorporating numerical features of the review, reviewer, and hotel helps classifiers obtain better performance, as shown by the findings from both Yelp_all and Yelp_balanced datasets. But questions like how much contribution each group of features

offers to improve the classifier's performance, have no answers in this study.

5 Conclusions and Future Research

Since the surge of e-commerce in the 2000s, online reviews have been studied by researchers from multiple disciplines such as computer science, marketing, psychology, and management. Shopping, posting own reviews, and reading reviews from others is becoming a part of our daily life. Meanwhile, everyone wants to avoid getting misled by fake reviews. One effective and recognized way is to develop a reliable automatic classification system to identify fake reviews. In this paper, the author compares the performance between various classifiers, across different algorithms, sets of features, and datasets.

This study starts by reviewing the widespread dissemination, motive, and impact of fake reviews. Then the author discusses the theories behind the algorithms, features and performance evaluation metrics, and existing findings from previous studies. Various algorithms including Logistic Regression, ML methods like SVM, NB, and RF, and DL methods such as MLP, LSTM, and BERT are in the scope of discussion. After that, the author presents features extracted from reviews, reviewers, and hotel information and metrics from accuracy, F1score, macro-F1score to AUC. Then, the author introduces four datasets to train classifiers: two are adopted and the other two are newly mined. From the datasets some combinations of pseudo/real, big/small, and balanced/imbalanced can be seen. Subsequently, the author tries different combinations of algorithm and features to develop various classifiers, trains on each dataset, and gets results of performance evaluation metrics then analyzes. Finally, the author draws findings from reading results across datasets.

Several interesting findings emerge from the analyses. The author describes them in subsection 4.2 and discusses in 4.3. Overall, results analysis suggests that DL algorithms have

good potential to obtain much better performance in contrast to ML algorithms, especially when the training sample consists of actual and class-imbalanced reviews, and various features are included as inputs. The Word Embeddings, either from scratch or pre-trained, helps relatively more to increase classifier's performance, than the one-hot text features, like word counts and TF-IDF.

This paper is one of the first that compares fake review identification performance of various classifiers, which employ algorithms including multiple ML and DL methods, train on different combinations of various features, and across a few datasets of different size and class balance condition.

5.1 Contributions

Theoretically, this study enriches the literature of two channels: the application of advanced DL algorithms to text-centered binary classification tasks, and the comparative research on different algorithms and features in fake review detection classification. In the marketing area, Berger et al. point out, development in DL and NLP-based approaches should enable marketing researchers to catch deeper textual relationships, which are beyond the plain co-occurrence of words and are more interesting to researchers(Berger et al., 2020). The authors also expect to see transfer learning methods can be taken in marketing to capture more complex behavioral states from the text. The pre-trained BERT model application in this study is a small step on this road. Marketing scholars from MIT and North Western, Urban et al. claim that although only marginal gains are achieved with old data, they still think the prospects for DL are bright in some contexts, especially as new DL-based algorithms and more comprehensive databases become available(Urban et al., 2020). In the computer science area, there is not much literature in which more than five algorithms, DL and ML algorithms

are simultaneously employed, more than three datasets are trained on, and multiple groups of features are used as inputs. This study is one of the first that attempts to do so, by collecting more findings, to evaluate models more objectively.

Empirically, the contribution of this study is three-fold. First, it offers some evidence that employing DL algorithms lead to better fake review detection performance than ML algorithms, especially in real-life contexts. Second, this study is also a preliminary attempt to see how beneficial it could be to use a BERT model for a fake review classification task, shedding some light on the road, though the available computational resource limits this study to some extent. Third, this study shows including features from not only the review itself but also information about the reviewer and product leads to better performance in most cases. Thus, this study shows a wonderful vision for researchers to collect more features for better identifying fake reviews.

Managerially, this study is a good trial that offers data and ideas to practitioners. It applies not only to online review website operators and business owners, but also to review viewers and law enforcement. It illustrates how to build up an automatic fake review classification model, by using mainstream algorithms and publicly accessible features, to detection evaluation metrics and how to compare performance across models in different contexts. This study also shows that though one algorithm may not always be the best across different contexts, putting efforts into studying how to well use of new and powerful DL algorithms is worthwhile and promising. Fully fine-tuned DL algorithms may achieve even better fake review identification performance. Furthermore, considering how hard it is for researchers to collect labeled dataset, this study may encourage the information sharing between researchers and practitioners.

5.2 Limitations

There are several limitations to this study. First, because of limited hardware resources, the author applies the BERT model to only one dataset using a very shallow network and a small portion of the data, so that the result of the 'fine-tuned' model is not that fine and should be better than what reports. Second, the study does not fully make use of all available information from the Yelp_Liu dataset. To keep the data structure of the four datasets similar and to make the study viable, when using the Yelp_Liu dataset, the author does not include abundant reviewer information offered by the original dataset. When manually mining the new dataset from Yelp, it is unmanageable to collect the same level of data as the original Yelp_Liu dataset, which was scraped over 8 years ago, without a high-performance GPU computer dedicated to crawling and processing data. Furthermore, again due to hardware limitations, even if more data is collected and ready, it still is not manageable for the author to train good DL models with enough deep network on the extended data. Third, grid search for the best hyperparameter has not been done in this study. Otherwise, the experimental findings may be more general. The author thinks since the aim of this study is not to train out a best fake review classification model, it is not entirely necessary to invest in doing a time-consuming but not decisive grid search job. But grid search itself is worthy of trying by obtaining best hyperparameters and accordingly, better classifier's performance given enormous computational resource is available.

5.3 Future research

Based on the current study, improvements from many directions can be made in the future.

Firstly, with a more powerful computational resource, DL algorithms can be made deeper with more hidden layers and be trained on a much larger dataset to make good use of their born

advantages. For instance, in the current study, the base pre-trained BERT word embeddings version is chosen purely because of its smaller size and fine-tuned on only 20% of the sample due to hardware limitations. Future work can try to use a larger dataset and combining BERT word embeddings and other features related to review, reviewer, and product as multiple inputs to the model. It looks promising for a better fake review identification BERT model to come out. Maybe until then, researchers will be more confident telling if employing DL algorithms leads to leapfrog or just improve slightly on fake review detection tasks.

Secondly, collect more features about the reviewer's behavior and use them as inputs. Previous studies (Mukherjee et al., 2013) have shown that behavioral features are important to identify fake review writers and reviews they post. Features can be extended from what has been suggested like the maximum number of reviews in a day, review polarity, review rating deviation, and review content similarity. Also, the increasingly mature computer vision technology helps fake review identification researchers to draw information from characteristics related to pictures such as whether pictures are used in a review, the percentage of reviews with pictures, the number of pictures in a review, whether the picture matches the product, and others. Similar improvements can apply to product feature collecting.

Thirdly, future research may also consider collecting and studying a dataset that includes ground-truth fake and truthful reviews, and training and evaluating classifiers with DL and ML algorithms. He et al. (2020) hand collect data on social media sites like Facebook and others, through which fake product reviews on Amazon.com are purchased, to characterize the types of products that buy fake reviews. In this way, worries about the reliability of the reviews' fake labels disappear. On the other hand, collecting real non-fake reviews may not be as reliable as the above, but it is still possible to collect reviews that are real non-fake that are

truthful.

Fourthly, the fine-tuning job can be carried out in-depth on various algorithms to get the best from each classifier. In contrast, the number of hyperparameters for DL algorithms is usually much larger than that for ML algorithms. With more powerful hardware resources, the fine-tuning job gets more fully done, studies can show more confidence in which algorithm leads to better performance after comparing the values of appropriate metrics between algorithms.

Fifthly, work on the application of a good classifier, such as use the classifier to predict the fakeness of new instance of online review. Then, further study the relationship between online reviews and managerial responses by splitting reviews into two classes, fake and non-fake, and may draw insights about the difference between the above relationship across the two classes.

Lastly, the same frame and method of this study can extend from hospitality to other industries, like medical service and others. In this way, we can get a much broader picture of how to identify fake online reviews in different contexts.

References

001-intro.pdf. (2020). Google Docs. Retrieved December 29, 2020, from https://drive.google.com/file/d/1Q7LtZyIS1f3TfeTGll3aDtWygh3GAfCb/view?usp=drive_open&usp=embed_facebook

Anderson, E. T., & Simester, D. I. (2014). Reviews without a Purchase: Low Ratings, Loyal Customers, and Deception: *Journal of Marketing Research*. https://doi.org/10.1509/jmr.13.0209

Barbado, R., Araque, O., & Iglesias, C. A. (2019). A framework for fake review detection in online consumer electronics retailers. *Information Processing & Management*, *56*(4), 1234–1244. https://doi.org/10.1016/j.ipm.2019.03.002

Barsever, D., Singh, S., & Neftci, E. (2020). Building a Better Lie Detector with BERT: The Difference Between Truth and Lies. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. https://doi.org/10.1109/IJCNN48605.2020.9206937

Berger, J., Humphreys, A., Ludwig, S., Moe, W. W., Netzer, O., & Schweidel, D. A. (2020). Uniting the Tribes: Using Text for Marketing Insight. *Journal of Marketing*, *84*(1), 1–25. https://doi.org/10.1177/0022242919873106

Black Friday on Amazon: How to spot fake reviews online. (2020, November 27). *BBC News*. https://www.bbc.com/news/newsbeat-54885319

Cortes, C., & Mohri, M. (2003). *AUC Optimization vs. Error Rate Minimization*. *3*.

Deceptive Opinion Spam Corpus Myle Ott. (n.d.). Myle Ott. Retrieved March 5, 2021, from https://myleott.com/op-spam.html

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Elmurngi, E., & Gherbi, A. (2017). *An empirical study on detecting fake reviews using machine learning techniques* (p. 114). https://doi.org/10.1109/INTECH.2017.8102442

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38. https://doi.org/10.1016/j.patrec.2008.08.010

Five-star fraud: Fake reviews are coming for your wallet - Trustpilot Blog. (2020). Trustpilot. Retrieved February 25, 2021, from https://www.trustpilot.com/blog/trends-in-trust/five-star-fraud-fake-reviews-are-coming-for-your-wallet

FTC Finalizes Settlement in LendEDU Case Related to Deceptive Rankings and Fake Reviews. (2020, May 26). Federal Trade Commission. https://www.ftc.gov/news-events/press-releases/2020/05/ftc-finalizes-settlement-lendedu-case-related-deceptive-rankings

Harmon, A. (2004, February 14). Amazon Glitch Unmasks War Of Reviewers (Published 2004). *The New York Times*. https://www.nytimes.com/2004/02/14/us/amazon-glitch-unmasks-war-of-reviewers.html

He, S., Hollenbeck, B., & Proserpio, D. (2020). *The Market for Fake Reviews* (SSRN Scholarly Paper ID 3664992). Social Science Research Network. https://doi.org/10.2139/ssrn.3664992

Hendrycks, D., & Gimpel, K. (2020). Gaussian Error Linear Units (GELUs). *ArXiv:1606.08415 [Cs]*. http://arxiv.org/abs/1606.08415

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735. https://doi.org/10.1162/neco.1997.9.8.1735

Inc, Y., & Monday, O. 5. (2009, October 5). *Why Yelp Has A Review Filter*. Yelp. https://blog.yelp.com/2009/10/why-yelp-has-a-review-filter

Jain, N., Kumar, A., Singh, S., Singh, C., & Tripathi, S. (2019). Deceptive Reviews Detection Using Deep Learning Techniques. In E. Métais, F. Meziane, S. Vadera, V. Sugumaran, & M. Saraee (Eds.), *Natural Language Processing and Information Systems* (pp. 79–91). Springer International Publishing. https://doi.org/10.1007/978-3-030-23281-8_7

Jawahar, G., Abdul-Mageed, M., & Lakshmanan, L. V. S. (2020). Automatic Detection of Machine Generated Text: A Critical Survey. *ArXiv:2011.01314 [Cs]*. http://arxiv.org/abs/2011.01314

Jindal, N., & Liu, B. (2008). Opinion spam and analysis. *Proceedings of the International Conference on Web Search and Web Data Mining - WSDM '08*, 219. https://doi.org/10.1145/1341531.1341560

Kennedy, S., Walsh, N., Sloka, K., McCarren, A., & Foster, J. (2019). Fact or Factitious? Contextualized Opinion Spam Detection. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 344–350. https://doi.org/10.18653/v1/P19-2048

Kumar, J. (2020). Fake Review Detection Using Behavioral and Contextual Features. *ArXiv:2003.00807 [Cs]*. http://arxiv.org/abs/2003.00807

Kumar, N., Venugopal, D., Qiu, L., & Kumar, S. (2018). Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning. *Journal of Management Information Systems*, *35*(1), 350. https://doi.org/10.1080/07421222.2018.1440758

Kumar, N., Venugopal, D., Qiu, L., & Kumar, S. (2019). Detecting Anomalous Online Reviewers: An Unsupervised Approach Using Mixture Models. *Journal of Management Information Systems*, *36*(4), 1313. https://doi.org/10.1080/07421222.2019.1661089

Lappas, T., Sabnis, G., & Valkanas, G. (2016). The Impact of Fake Reviews on Online Visibility: A Vulnerability Assessment of the Hotel Industry. *Information Systems Research*, *27*(4), 940–961. https://doi.org/10.1287/isre.2016.0674

LeCun, Y. (2019). 1.1 Deep Learning Hardware: Past, Present, and Future. *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, 12–19. https://doi.org/10.1109/ISSCC.2019.8662396

LeCun, Yann, Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lewis, G., & Zervas, G. (2019). *The Supply and Demand Effects of Review Platforms*. Available at SSRN: https://ssrn.com/abstract=3468278 or http://dx.doi.org/10.2139/ssrn.3468278

Li, H., Liu, B., Mukherjee, A., & Shao, J. (2014). *Spotting Fake Reviews using Positive-Unlabeled Learning*. Comutacion Sistemas, *18*(3), 467-75.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, 2(3), 18-22.

Local Consumer Review Survey: How Customer Reviews Affect Behavior. (2020, December 9). BrightLocal. https://www.brightlocal.com/research/local-consumer-review-survey/

Luca, M. (2016). *Reviews, Reputation, and Revenue: The Case of Yelp.Com* (SSRN Scholarly Paper ID 1928601). Social Science Research Network. https://doi.org/10.2139/ssrn.1928601

Luca, M., & Zervas, G. (2016). Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud. *Management Science*, *62*(12), 3412–3427. https://doi.org/10.1287/mnsc.2015.2304

Malik, F. (2019, May 18). *Understanding Neural Network Neurons*. Medium. https://medium.com/fintechexplained/understanding-neural-network-neurons-55e0ddfa87c6

Mayzlin, D., Dover, Y., & Chevalier, J. (2014). Promotional Reviews: An Empirical Investigation of Online Review Manipulation. *American Economic Review*, *104*(8), 2421–2455. https://doi.org/10.1257/aer.104.8.2421

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. https://arxiv.org/abs/1310.4546v1

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., & Ghosh, R. (2013). Spotting opinion spammers using behavioral footprints. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '13*, 632. https://doi.org/10.1145/2487575.2487580

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013). *Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews*.11.

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning.* https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf

Proceedings of the 27 th International Conference on Machine Learning

Opitz, J., & Burst, S. (2019). Macro F1 and Macro F1. *ArXiv:1911.03347 [Cs, Stat]*. http://arxiv.org/abs/1911.03347

Ott, M., Cardie, C., & Hancock, J. (2012). Estimating the prevalence of deception in online review communities. *Proceedings of the 21st International Conference on World Wide Web - WWW '12*, 201–210. https://doi.org/10.1145/2187836.2187864

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative Deceptive Opinion Spam. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 497–501. https://www.aclweb.org/anthology/N13-1053

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding Deceptive Opinion Spam by Any Stretch of the Imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 309–319. https://www.aclweb.org/anthology/P11-1032

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The Development and Psychometric Properties of LIWC2015*. https://doi.org/10.15781/T29G6Z

Raval, D. (2020). *Do Bad Businesses Get Good Reviews? Evidence from Online Review Platforms*. 51.

Rayana, S., & Akoglu, L. (2015). Collective Opinion Spam Detection: Bridging Review Networks and Metadata. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, 985–994. https://doi.org/10.1145/2783258.2783370

Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, *385–386*, 213–224. https://doi.org/10.1016/j.ins.2017.01.015

Rout, J. K., Dalmia, A., Choo, K. R., Bakshi, S., & Jena, S. K. (2017). Revisiting Semi-Supervised Learning for Online Deceptive Review Detection. *IEEE Access*, *5*, 1319–1327. https://doi.org/10.1109/ACCESS.2017.2655032

Ruan, N., Deng, R., & Su, C. (2020). GADM: Manual fake review detection for O2O commercial platforms. *Computers & Security*, *88*, 101657. https://doi.org/10.1016/j.cose.2019.101657

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Shahariar, G. M., Biswas, S., Omar, F., Shah, F. M., & Hassan, S. B. (2019). Spam Review Detection Using Deep Learning. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 0027–0033. https://doi.org/10.1109/IEMCON.2019.8936148

Shukla, A., Wang, W., Gao, G. (Gordon), & Agarwal, R. (2019). *Catch Me If You Can—Detecting Fraudulent Online Reviews of Doctors Using Deep Learning* (SSRN Scholarly Paper ID 3320258). Social Science Research Network. https://doi.org/10.2139/ssrn.3320258

Sun, C., Du, Q., & Tian, G. (2016, August 4). *Exploiting Product Related Review Features for Fake Review Detection* [Research Article]. Mathematical Problems in Engineering; Hindawi. https://doi.org/10.1155/2016/4935792

Tang, J., Yang, Y., Carton, S., Zhang, M., & Mei, Q. (2016). Context-aware Natural Language Generation with Recurrent Neural Networks. *ArXiv:1611.09900 [Cs]*. http://arxiv.org/abs/1611.09900

Urban, G., Timoshenko, A., Dhillon, P., & Hauser, J. R. (2020). Is Deep Learning a Game Changer for Marketing Analytics? *MIT Sloan Management Review; Cambridge*, *61*(2), 70–76.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv:1706.03762 [Cs]*. http://arxiv.org/abs/1706.03762

Why would a review not be recommended? | Support Center | Yelp. (n.d.). Retrieved January 4, 2021, from https://www.yelp-support.com/article/Why-would-a-review-not-be-recommended?l=en_US

Wu, X., Dong, Y., Tao, J., Huang, C., & Chawla, N. V. (2017). Reliable fake review detection via modeling temporal and behavioral patterns. *2017 IEEE International Conference on Big Data (Big Data)*, 494–499. https://doi.org/10.1109/BigData.2017.8257963

Yelp—Company—Fast Facts. (2021). Retrieved January 5, 2021, from https://www.yelp-press.com/company/fast-facts/default.aspx

Zhao, S., Xu, Z., Liu, L., Guo, M., & Yun, J. (2018). Towards Accurate Deceptive Opinions Detection Based on Word Order-Preserving CNN. *Mathematical Problems in Engineering*, *2018*, 1–9. https://doi.org/10.1155/2018/2410206

Chapter/Essay II

Online Reviews, Managerial Responses, and Hotel Ratings:

Evidence from Tripadvisor

Abstract

The purpose of this study is twofold. First, to explore factors contributing to the likelihood of a review to receive a managerial response (MR), testing the impact of the fake review detection results, review congruency, review deviation, and the moderating role of hotel class. Second, to examine the association among online reviews, managerial responses, and hotel rating (growth rate), including both text similarity and fake review detection results as independent variables. This study is one of the first that introduces fake review detection and text similarity into research about MR, adding to the literature of MR in the context of Tripadvisor.com. Our findings indicate the following practical implications. (1) A truthful, detailed, and congruent review is more likely to receive an MR; (2) The percentage of truthful reviews has a strong and positive association with hotel rating and its growth. In an extreme situation, the hotel rating will go up by 0.21, and the rating growth rate will increase by 8.5% due to 100% truthful reviews; (3) Hotels should carefully choose which review(er)to respond to and make responses concise and matching while actively responding to reviews.

Keywords: Fake review detection; Online reviews; Managerial responses; Hotel ratings

1 Introduction

On major travel websites, both hotel consumers and hoteliers have room to voice publicly. People post reviews about their experience, complaints, or satisfaction, whereas managers reply to reviews and express empathy or gratitude. Though hotels usually have adopted the response strategy to accommodate the economic impact of electronic word of mouth (eWOM) (Babić Rosario et al., 2016; Chevalier & Mayzlin, 2006; Luca, 2016), not every review receives its corresponding managerial response (MR) because of a limited budget or other considerations.

One major stream of research is on the externality on the opinions of review readers, proactive consumers, and subsequent review and reviewers (Chen et al., 2019; Proserpio & Zervas, 2017; Y. Wang & Chaudhry, 2018; K. L. Xie et al., 2016), and the internality on the returning consumers (Gu & Ye, 2014; Y. Xu, Li, et al., 2020; Zhao et al., 2020). Recent studies further find, arguably, the positive effect of MR on hotel performance. Furthermore, given the well-documented benefit from MR strategy and limited resources to respond, a few studies examined what kind of MR strategy is most beneficial to hotels. The studies involve multiple aspects of MR strategy, including which or whom to target, time zone, and text style (Hogreve et al., 2017; Liang & Li, 2019; L. Wang et al., 2020; X. Zhang et al., 2020; Z. Zhang et al., 2019).

Another research stream has focused on suspicious review and its impact on ratings and sales across multiple industries (Anderson & Simester, 2014; He et al., 2020; Luca, 2016; Luca & Zervas, 2016; Mayzlin et al., 2014; Y. Xu, Zhang, et al., 2020; Zheng et al., 2021a). Suspicious or even fake reviews have increased in volume and been widely seen on review platforms and e-commerce sites, as businesses intend to stand out in contrast to their competitors and commit review manipulation as a part of their online reputation management strategies (Luca, 2016; Luca & Zervas, 2016; Mayzlin et al., 2014). Anderson & Simester (2014) states that deceptive reviews may be beyond the strategic actions of firms and extend to individual customers who want to influence product ratings.

There is little overlap between these two streams. For instance, we know little about the connection between suspicious reviews and MRs. Surachartkumtonkun et al. (2020) are the first to study how to employ MR strategy to minimize the negative influence of deceptive reviews, from the perspective of a third-party in an experimental context. From a different aspect from above, this study takes a deep learning fake review classifier's predictive outcome as labels that indicate the fakeness of reviews. This study then examines the connection between suspicious reviews and MRs in a more straightforward way.

We study hotel reviews, MRs, and ratings on Tripadvisor.com to document the association among the three by applying fake review detection and text analysis methods. First, we use a logistic regression model to capture the determinants of the likelihood for a review to receive an MR. In line with congruity theory, we include congruency, which is an indicator showing whether a review's rating falls in the same category as hotel rating; and deviation, which measures the distance between review rating and hotel rating. The model also includes the Fake or NFake label, which is attached to a review as the classification result of a fake review classifier. In addition, we testify to the moderating role of hotel class. Second, based on the previous study, we further examine the interrelationship between dependent variables, hotel rating and its growth, and independent variables, fake review detection results and three attributes of MR strategy, at an aggregated hotel-month level.

Regarding the first research question, we found a truthful (fake) and congruent (incongruent) review with some deviation is more (less) likely to receive an MR, and the effect is attenuated if the focal review is about a top-class hotel. For the second question, we found significant connections between hotel rating/growth and average truthful level (positive), MR ratio level (positive), MR length (negative), and text similarity (positive), but we caught no negative relation for average fake level factor. Furthermore, we found that for hotels that have regularly responded to reviews, the MR ratio level does not significantly relate to hotel rating and its growth. Our empirical findings guide

how to better understand the selectivity of MR strategy and effectively use MRs to influence the valence of eWOM.

Our study contributes to the literature on eWOM in a few ways. First, we build a bridge to connect the two aforementioned literature streams, by using the predictive classification results as variables of interest. Second, this study adds to the growing literature which embraces the advancement of computer science technology in marketing research (Berger et al., 2020; Ma & Sun, 2020; Urban et al., 2020; Y. Wang & Chaudhry, 2015). We show textual analysis (text similarity) and a deep learning approach (fake review classifier) can be applied to draw out good indicative meanings for both consumers and marketers.

The article proceeds as follows. Section 2 reviews relevant literature and proposes a set of hypotheses on the association among fake review detection, managerial response, and hotel rating. Section 3 describes research methodology and data. Section 4 presents the findings and results of hypothesis testing. Section 5 discusses our contributions, implications, and limitations.

2 Literature and hypotheses

This study focuses on two topics: (1) online review and MR; (2) the joint effect of online review and MR on hotel overall rating and its growth. We discuss these areas of study as follows.

2.1 Online review and managerial response

When exploring the relationship between reviews and responses, most previous literature use features of MR as independent variables, assign the valence, volume, and variance of subsequent reviews as dependent variables (Chen et al., 2019; Proserpio & Zervas, 2017; Y. Wang & Chaudhry, 2018), since MR strategy is a tool to serve eWOM by maintaining good

public relationship. This research intends to explore the connection from an opposite angle, that is, what kind of review should get an MR to make the MR strategy a successful reputation management tool, and whether and how the reality echoes our findings.

2.1.1 Fake review identification

To label a review as fake or truthful, this study harnesses a pre-trained fake review classifier, which is trained on a large, labeled sample dataset from Yelp.com, proven to get 85% accuracy and outperform other classifiers.

Suspicious reviews maybe more prevalent than expected. Previous studies show that the share of fake reviews is about 15-30% (Lappas et al., 2016; Luca & Zervas, 2016), despite the filtering system efforts of review platforms make (He et al., 2020). Anderson and Simester (2014) point out that usual customers are unlikely to identify deceptive reviews. Given multiple sources of fake reviews, on one hand, platform or hotel can pinpoint a large portion of reviews as fake or truthful with high confidence when the origin of reviews is clear. For instance, a hotel that commits positive review fraud for itself knows which positive review is purchased or manufactured by itself, or a hotel that catches negative review fraud by its competitors knows which negative review is purchased or manufactured by its competitors. In terms of a platform, its identification system helps it catch suspicious reviews. On the other hand, a platform or hotel may not have high confidence in identifying if a review is fake or truthful when the origin of the review is unknown, as Anderson and Simester (2014) suggest. So, this low confidence may necessitate a third category, undecided, between fake and truthful.

For positive untruthful reviews which are made or purchased by the focal hotel itself, they aim to boost the overall hotel rating instead of getting attention from customers; for those negative untruthful reviews coming from competitors or from other individuals, the review

platform will probably remove them after the hotel appeals to, so managers are inclined not to respond. In addition, according to service failure theory (Spreng et al., 1995) and affective theory (Groth et al., 2009), hotels should be more likely to respond to truthful reviews, either positive or negative, since MR strategy aims to moderate consumer satisfaction by showing empathy or gratitude.

On the above basis, we propose the following hypotheses:

H1a. A review labeled as fake is less likely to receive an MR in contrast to an undecided one.

H1b. A review labeled as truthful is more likely to receive an MR in contrast to an undecided one.

2.1.2 Review Deviation

In this study, we define review deviation as the rating difference between individual review and the average of all reviews in a month. Deviation represents, to some extent, the interaction between individual review rating and other reviews in a focal period. Aral (2013) argues that rating bubbles and the "J-shaped distribution" of online ratings are widespread, and that the reason behind this is our herd instincts. In the framework of herd instincts, people are likely to agree with others, heavily right-skewed toward positive opinions. So, deviation measures the distance between individual review and the average of other reviews, showing a review's heterogeneity.

According to the signaling theory (Connelly et al., 2011), a review with a deviated rating serves as a signal, carrying some unique information. The larger the deviation of a review, the more additional information it bears. Combined with congruency—a binary indicating whether there is divergence between individual and population, deviation presents

the review's uniqueness by measuring the magnitude of difference. From the hotel's perspective, responding to a review with a high level of uniqueness will get more attention from readers, and thus either promote the hotel's positive image or address the problem to a larger size of audience. Therefore, hotel is more likely to make a response to the inconsistent information carried by reviews with more deviation.

We propose the hypothesis:

H1c. A review with more deviation is more likely to receive an MR, in contrast to a review with less.

2.1.3 Congruency

In this study, we define congruency of a review as an indicator showing whether the rating of an individual review falls in the same category as the monthly average review rating of the focal hotel in the focal month. There are two rating categories according to numerical value of ratings: above 4-star and under 4-star since many large-size secondary datasets across multiple hotel review or booking platforms show 4-star rating is the mean or median across all sample data (Chevalier et al., 2018; Luca & Zervas, 2016). This categorization agrees with many previous studies, which group reviews above 4-star as positive and under as negative. Looking at an individual review, people check whether it is about a good rating hotel or an under-average, and whether the rating of the review itself matches that of hotel or not. If a review is positive for an above-average hotel, the information is matched, congruent, and reinforced, while the information from an unmatched review is incongruent and contradicted.

From a theoretic perspective, congruity theory (Osgood & Tannenbaum, 1955) suggests that a congruent review is more likely to be agreed by proactive consumers or review readers, as compared to an incongruent one. For the instance of a positive review for an above-average

hotel, people tend to agree with it because they do not feel pressure to reconcile these two messages, while in the instance of a positive review for an under-average hotel, people may feel pressure to reconcile when confronting the incongruent messages and thus are less likely to agree with it. Similarly, in the instances of negative review, people tend to agree when it is for an under-average hotel rather than an above-average hotel. Attribution theory (Folkes, 1988; Kelley & Michela, 1980) suggests that people are more likely to attribute the incongruent reviews to the reviewer (poster) rather than the hotel. So, responding to a positive review will reinforce favorable information for an above-average hotel at no communication cost, but may cause more inconsistent information for an under-average hotel.

Regarding responding to a negative review, attribution theory only looks unapplicable since it is known that the negative review is more influential to eWOM. Service failure recovery theory (McCollough et al., 2000) argues timely responding to negative review is the first step to help hotel reputation get recovered from service failure and a chance to rebuild hotel image. With this perspective, managers should respond. In the context of an under-average hotel, high percentage of responding to negative reviews is well understood from a combined perspective originated from the attribute theory and service failure recovery theory. Whereas in the context of an above-average hotel, managers may not rush to respond. A few studies show that responding more and in more detail to negative response may stimulate negative reviewing activity (Chevalier et al., 2018), and a fellow consumer response to a negative review benefit the hotel the most (Esmark Jones et al., 2018). Given positive reviews are prevalent, an above-average hotel may not need to respond to all negative reviews, which only account for a small portion. So, responding to a negative review offers an under-average hotel an opportunity to address the issue, clear its name, and recover from its service failure, but may lead to more negative reviews for an above-average hotel.

Based on above, we propose the following hypothesis:

H1d. A congruent review is more likely to receive an MR in contrast to an incongruent review.

In other words, responding to a positive review for an above-average hotel and to a negative review for an under-average hotel are more likely than to a negative for an above and to a positive for an under, respectively.

2.1.4 Moderating role of hotel class

People categorize hotels into segments such as economic or premium, depending on the star ranking. The star ranking of a hotel, from the lowest 1-star to the highest 5-star, shows its quality and position in the industry and among consumers (Israeli, 2002). Usually, hotels with 4-star or above, in this study, we call them as top-class hotels, are at first-class or luxury level, with an excellent reputation. As compared to others, top-class hotels have a bigger brand name and are also more familiar to people. Therefore, prospective tourists rely on reviews from others at a much lower level since the information from other sources about the focal hotel is already abundant and accessible. Similarly, tourists do not rely on MR that much to infer the quality of top-class hotels. In addition, previous studies (Y. Wang & Chaudhry, 2018) show that people consider (may inaccurately) responding to positive reviews as a promotional activity. Considering top-class hotels have well above-average review ratings, the hypothetical positive correlation between congruency and the chance of receiving an MR, as stated in H1d, should be attenuated.

Thus, we propose the following hypothesis:

H1e. The association between the congruency of a review and the chance of receiving an MR is stronger for non-top-class hotels than for others.

2.2 Fake review detection, managerial response, and hotel rating

Many previous studies have found the positive impact of MR on subsequent review ratings with significance. But as we mentioned earlier, almost all studies did not examine the association of fake review identification with hotel rating. This study is one of the first to make the attempt.

2.2.1 Fake review detection

The distribution of fake review rating presents a U shape, indicating extreme negative and positive accounts for a larger portion. High fake level means that either managers proactively and continuously manipulate the eWOM, or the hotel gets heavily attacked by competitors, or get many false reviews from entities who want to influence the hotel's eWOM. On the contrary, high truthful level means that fewer review manipulation activities.

He et al. (2020) found only one month after firms stop buying fake 5-star reviews via social media platform their ratings decrease abruptly. There is no study capturing fake 1-star reviews purchase and posting, and thus no empirical evidence on the correlation between fake negative review level and ratings.

In terms of high truthful level, it represents high level of trustiness of the hotel and review. Previous studies show the positive effect of trustiness on ratings. Considering the relatively opposite influence from fake reviews and truthful reviews, we propose the following hypotheses:

H2/3a. The level of fake review proportion negatively associates with the level and the growth rate of the ratings of consumer eWOM.

H2/3b. The level of non-fake review proportion positively associates with the level and the growth rate of the ratings of consumer eWOM.

2.2.2 Managerial response

In this study, we are interested in the association between the valence of overall hotel rating and three major aspects of MR strategy, including MR ratio, MR length, and text similarity between review and response, on the condition of fake and truthful review level. We also care about the association between the increase in overall hotel rating valence and above variables.

Three focal facets of MR strategy denote during a period the average level of MR relative frequencies (MR ratio), number of words in each MR (MR word count), and textual similarity between a pair of review and response (Text similarity). One stream of research agrees on the significant positive causal effect of adopting MR strategy on subsequent eWOM, valence, and volume (Chen et al., 2019; Chevalier et al., 2018; Proserpio & Zervas, 2017; Y. Wang & Chaudhry, 2018). Another stream of research explores the correlation at an aggregated level between MR strategy and subsequent eWOM, showing relatively diversified findings. This study adds to research of the latter type, focusing on the association, not the causal effect.

2.2.2.3 MR ratio

Chevalier et al. (2018) shows that detailed response (a lengthy response) to negative reviews may lead to stimulate negative reviews and thus increase the share of negative reviews, while Proserpio & Zervas (2017) argue the presence of MRs may increase the cost of reviewer who posts negative reviews and thus reduce the share of negative reviews. Y. Wang & Chaudhry (2018) reconcile the contradict in the findings from the above two studies by dividing MRs to MP-Ns (to negative reviews) and MP-Ps (to positive reviews). Their study further points out that to obtain the largest possible positive externality of MR strategy on subsequent ratings, managers should customize MR-Ns to each review and limit the tailoring of MR-Ps.

In addition, consumers probably interpret every response to review as proof of empathy (to negative review), caring, responsible, willing to improve, and being helpful, showing a positive brand image. A study (Xu, Li, et al., 2020) shows a reviewer receiving an MR may

form a sense of reciprocity and want to pay back. So, a proactive consumer when seeing MR may establish the sense of reciprocity and be more kind to the focal hotel when posting her own review later. Therefore, we propose the following hypotheses about MR ratio:

H2/3c. The level of MR positively associates with the level and the growth rate of the ratings of eWOM.

2.2.2.3 MR length

Recalling the "J shape" of review rating distribution, we know that usually there are far more positive reviews than negative reviews. Given the findings from previous studies, it should be wise to respond to positive reviews in a shorter way but respond to negative reviews in a detailed way. So, when the average length of MR is long, there must be a relatively large share of negative reviews. Considering the herd instincts and bigger weight on recent information, lengthy MR should lead to a lower average rating. We propose the following hypotheses about MR length:

H2/3d. The length of MRs negatively associates with the level and the growth rate of the ratings of eWOM.

2.2.2.3 Text similarity

In terms of the style of MR, recent studies claim that responding from a tablet may be perceived as insincere, and making managerial response customized is a promising way (Y. Wang & Chaudhry, 2018; X. Zhang et al., 2020). But whether a customized response style is beneficial to rating level or financial performance and how to customize response still needs more evidence.

Y. Wang & Chaudhry (2018) calculate the cosine similarity between topic vectors of each review and its corresponding MR by using Latent Dirichlet Allocation (LDA) technology

to extract ten topics from nine million reviews and MRs. Min et al. (2015) find that empathy or paraphrasing response is helpful to increase prospective customers' satisfaction. Zhang et al. (2020) examine the influence of personalized managerial response on rating increase from a topic matching perspective. Z. Zhang et al. (2019) find managerial responses with high similarity can significantly reduce the hotel booking on Expedia. Li et al. (2017) capture the level of tailoring by using linguistic style matching (LSM) as a control variable when studying the effect of managerial response on subsequent consumer engagement. This study introduces a simple way to address the question: text similarity between review and response, using text analysis technology.

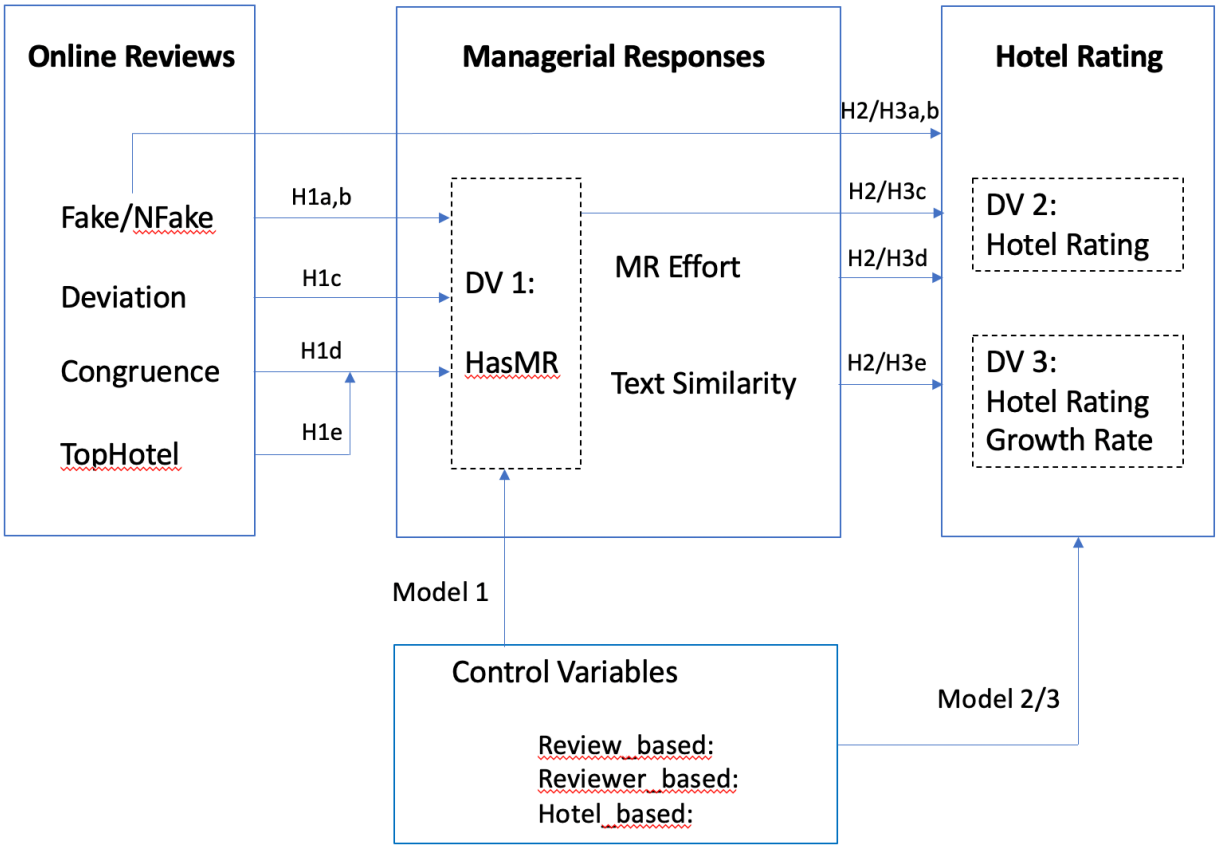
Based on previous findings, we propose the following hypotheses about text similarity:

H2/3e. The text similarity between MRs and reviews positively associates with the level and the growth rate of the ratings of eWOM.

2.3 Research framework

The research framework is shown in Figure 13.

Figure 13 Research framework



3 Methodology and data

3.1 Research context

The data of this study is from Tripadvisor.com, which was launched in the U.S. in 2000 by Tripadvisor, a leading online travel company worldwide. As of December 31, 2020, Tripadvisor featured 884 million reviews and opinions on 7.9 million hotels and other travel related accommodations (Tripadvisor, 2021). The website works as a platform for consumers to post comments of several kinds of forms on hotels and services, including reviews, photos, videos, and ratings, and for hotels to showcase and respond to reviews. Because of the affluent information about reviews and responses it offers, Tripadvisor has been one popular platform

for researchers to collect data, especially for those who study on the interaction between hotel online reviews and managerial response (Chang et al., 2020; Chevalier et al., 2018; Kwok & Xie, 2016; Proserpio & Zervas, 2017; Y. Wang & Chaudhry, 2018; K. L. Xie et al., 2016; K. L. Xie & So, 2018; X. Zhang et al., 2020).

3.2 Data

Like many prior studies, this study developed web crawlers to collect four aspects of data from Tripadvisor.com.: reviews, responses, reviewers, and hotels. We chose Las Vegas, Seattle, and San Francisco as the target areas and set the period from January 2017 to December 2019. For each review, we collected review text content, overall rating on a range of one to five, timing, and useful vote counts. We also recorded if it has a photo and the number of photos. If the review got a response from management, we extracted the MR text content, timing, and who responded. Besides, we recorded hotel class, which indicates the hotel level from 1 to 5 (1 means economical, 5 means luxury, 2, 3, and 4 stand in between). The initial dataset carried 458 hotels with 349,435 reviews after deleting non-English reviews or reviews with no text content. Considering one objective of this paper is to study the trigger for one review to get an MR, we further removed the reviews from hotels that did not respond during the period. In the end we got a dataset with a size of 346,617 reviews for 392 hotels.

Based on the data we got, we constructed two datasets with different units of analysis. The first is at the individual review and managerial response level, combining each review, response, and other related features. This dataset aims to examine the association between attributes of review such as its Fake/NFake tag, rating deviation, congruency, and the probability for it to receive a response, and the moderating role of hotel class. The second dataset is panel data, hotel-month, which was manually aggregated based on the first dataset.

The aggregated data comprises 13,081 hotel-monthly instances, involving 392 hotels and 36 months. The dataset aims to explore the effects of Fake/NFake level and three major attributes of managerial response on the hotel rating and its growth rate.

Figure 14 shows a screenshot of the webpage on Tripadvisor from which we collected data.

Figure 14 Screenshot of a hotel review and a managerial response from Tripadvisor.com

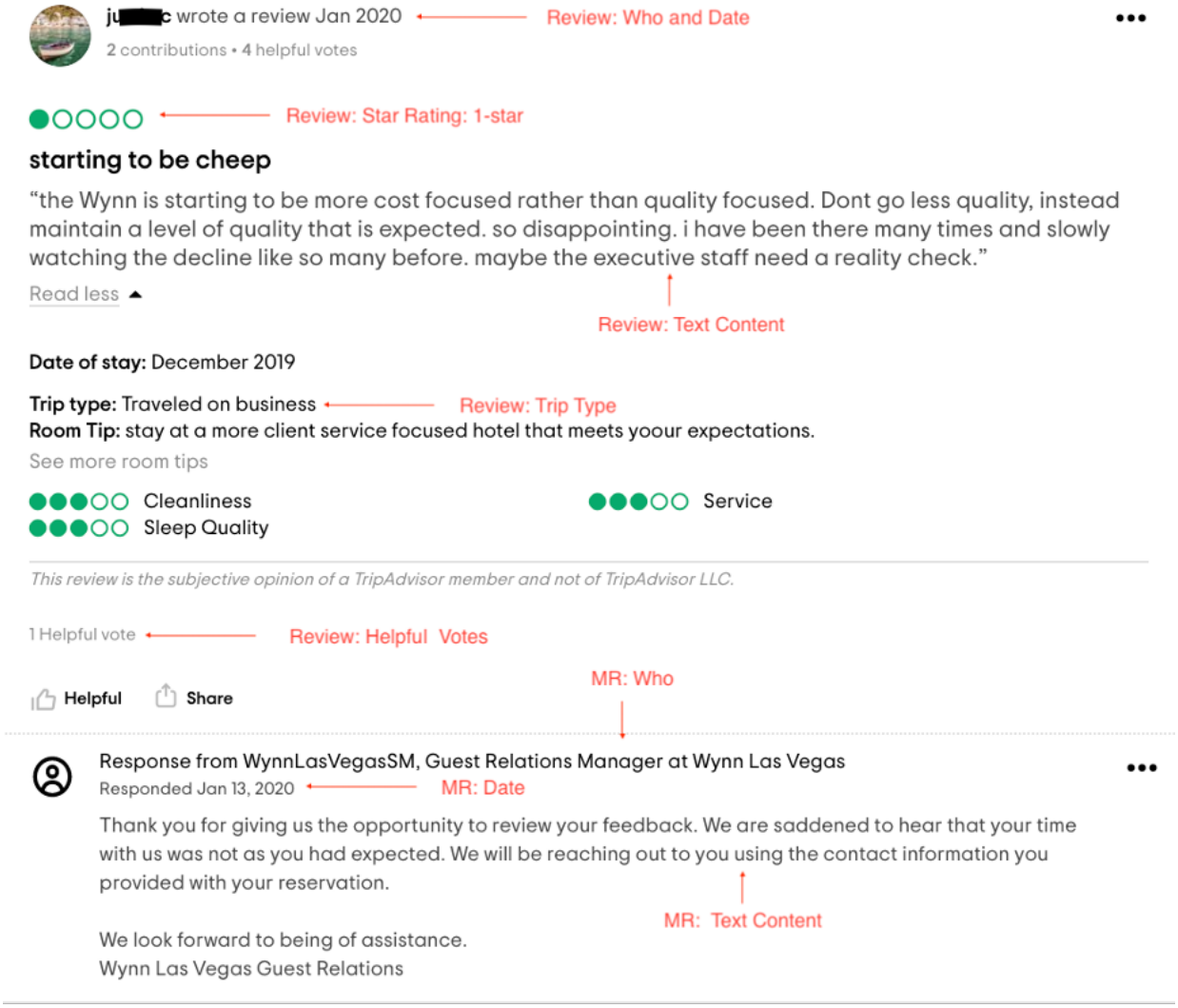


Table 12 provides descriptive statistics of the sample. Among 392 hotels, 154 are in Las Vegas, 82 in Seattle, and 156 in San Francisco. In terms of the share, hotels in Las Vegas account for 69.62% of reviews, followed by San Francisco (18.72%), and then Seattle (11.66%). In contrast, more than half reviews (1 - 47.17%) about hotels in Las Vegas did not

receive MRs, while 71.07% and 64.85% of the reviews about hotels in Seattle and San Francisco got MRs.

Table 12 Sample description based on city information

City	Number of Hotels	Number of Reviews	Pct of Reviews	Number of MRs	Average Class*	MR Ratio
Las Vegas	154	241,323	69.62%	113,833	3.26	47.17%
Seattle	82	40,424	11.66%	28,737	3.32	71.07%
San Francisco	156	64,870	18.72%	42,066	3.16	64.85%
Total	392	346,617	100%	184,636	3.23	53.27%

Notes: *The Hotel Class star ratings are provided from third-party partners, such as Giata, and national ratings organizations, indicating the general level to expect of features, amenities, and services offered by a hotel. We found the Hotel Class information on the hotel's listing page in the "About" section, according to Tripadvisor's guidance ("Updating My Hotel Star Rating.". The highest possible value is 5 for a luxury hotel, and the lowest possible value is 1 for a budget traveler hotel.

Table 13 presents descriptive statistics from the hotel class perspective. Among 392 hotels, 121 are of class 2.5 or under, 151 are of class 3 or above but under 4, and 120 are above class 4. The higher the class, the better service and facility are expected. Table 13 shows that different hotels do not have a sizable difference in the overall review ratings, although they are in distinct classes. The top-class hotels got the highest review ratings (4.13 out of 5) and the middle- and bottom-class hotels got very close review ratings, 3.83 and 3.79, respectively. In terms of the number of reviews, the top-class hotels provided the largest part (61%), followed by middle-class hotels, and only 6.77% of reviews were from the bottom-class hotels. For the MR ratio, we can see that the bottom-class hotels responded much more (66%) than the middle- and top-class hotels (52.67% and 52.17%, respectively).

Table 13 Sample description based on hotel class

Hotel Class	Review Rating	Number of Hotels	Number of Reviews	Pct of Reviews	Number of MRs	MR Ratio
Bottom_Class <= 2.5	3.79	121	23,459	6.77%	15,484	66.00%
Middle_Class >= 3 & < 4	3.83	151	110,674	31.93%	58,293	52.67%
Top_class >= 4	4.13	120	212,484	61.30%	110,859	52.17%
Total	4.01	392	346,617	100.00%	184,636	53.27%

Table 14 briefly states the number of reviews and MRs on each level of review ratings. Almost half the reviews, 49.07%, were 5-star, followed by 4-star (24.22%), 3-star (12.62%), and then 2-star, and 1-star combined only account for 14%. But when things come to the MRs, the order was almost the opposite. Table 14 shows 67.63% and 73.22% of 1-star and 2-star received MRs while only less than one half of 4-star (45.65%) and 5-star (49.18%) did.

Table 14 Sample description based on review rating

Review Rating	Number of Reviews	Percentage of Reviews	Number of MRs	MR Ratio
1	25,435	7.34%	17,201	67.63%
2	23,406	6.75%	17,138	73.22%
3	43,736	12.62%	28,322	64.76%
4	83,956	24.22%	38,328	45.65%
5	170,084	49.07%	83,647	49.18%
Total	346,617	100%	184,636	53.27%

Figure 15 presents the review rating distribution for three hotel class categories. As we can see, the rating distribution for each hotel class category looks like a J shape more or less. There are more 1-star reviews than 2-star, and the number of 3-, 4-, and 5-star reviews is more than one-star and shows a monotonically increasing trend. The steepest J shape is for top-class hotels, over 54% reviews are 5-star, 12% for 1-star and 2-star combined. For mid-class category, the shape is much smoother with 41% of 5-star and 17% for 1-star and 2-star. The shape for the bottom-class category is more like a cricket bat, with 37% of 5-star and 20% for 1-star and 2-star. Figure 16 shows a slowly decreasing trend in review rating for middle and above class hotels, while the review rating for the bottom-class hotels did not see a similar decreasing trend but a larger variance during 2017 - 2019.

Figure 15 Review rating distribution

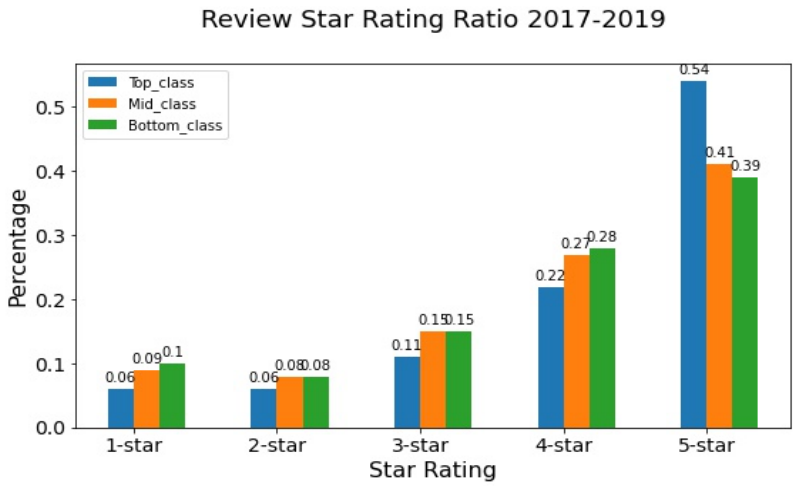


Figure 16 Review rating 2017-2019

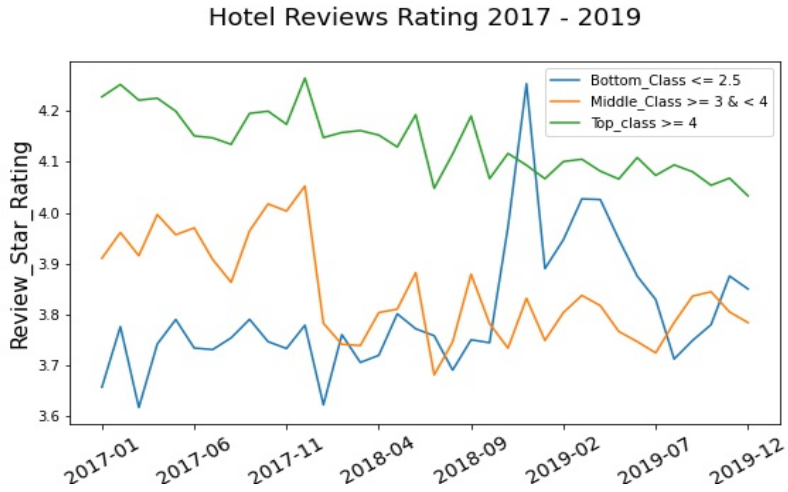


Figure 17 presents the distribution of MR ratio across hotel classes and review ratings. K. L. Xie et al. (2016) discovered in their sample data from Tripadvisor, retrieved from 2005 to 2011, that extremely positive, 5-star, and negative, 1-star ratings draw more attention from hotels, we found MR ratio presents differently across hotel classes. From Figure 17, we see for negative 1- and 2-star reviews, the top-class and the mid-class hotels responded over 66%, but positive 4- and 5-star reviews got MRs at a much lower rate, less than 49%. In contrast, bottom-class hotels responded with a higher probability and fairly evenly to each ratings level, over 61%, while hotels with higher classes made their responses differently: to the most positive 5-star reviews at the highest 71% and to the most negative 1-star at the lowest 61%.

We also noticed that from 2017 to 2019, the response strategy of hotels from different class categories had transformed. Figure 18 presents that, over the 36 months from 2017 to 2019, the middle and top-class hotels consistently decreased their MRs ratio across all rating levels, especially to the positive 4- and 5-star reviews. For instance, in Jan 2017, the top-class hotels responded to 1-star negative reviews at 83%, while in Dec 2019 sharply down to only 36%. In contrast, the bottom-class hotels responded to reviews with a relatively stable ratio, not showing the same decreasing trend as other hotels in the higher-class categories. Detailed monthly average information can be seen from Table 33 and Table 34 in the Appendix. This finding discloses that hotels, especially hotels in the middle and above class, have been reshaping their MR strategies, while they still kept responding to both negative reviews to make efforts trying to recover service failure and positive ones to address care and notice (Gu & Ye, 2014; Y. Wang & Chaudhry, 2018; K. L. Xie et al., 2016; K. L. Xie & So, 2018).

Figure 17 MRs ratio average over 2017-2019

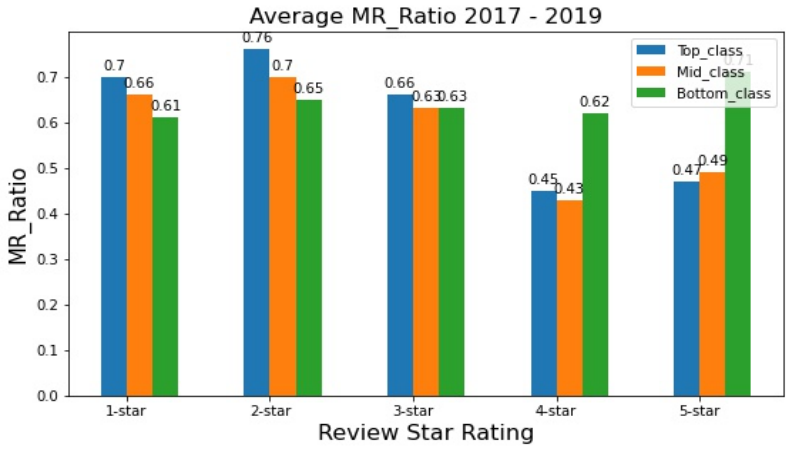
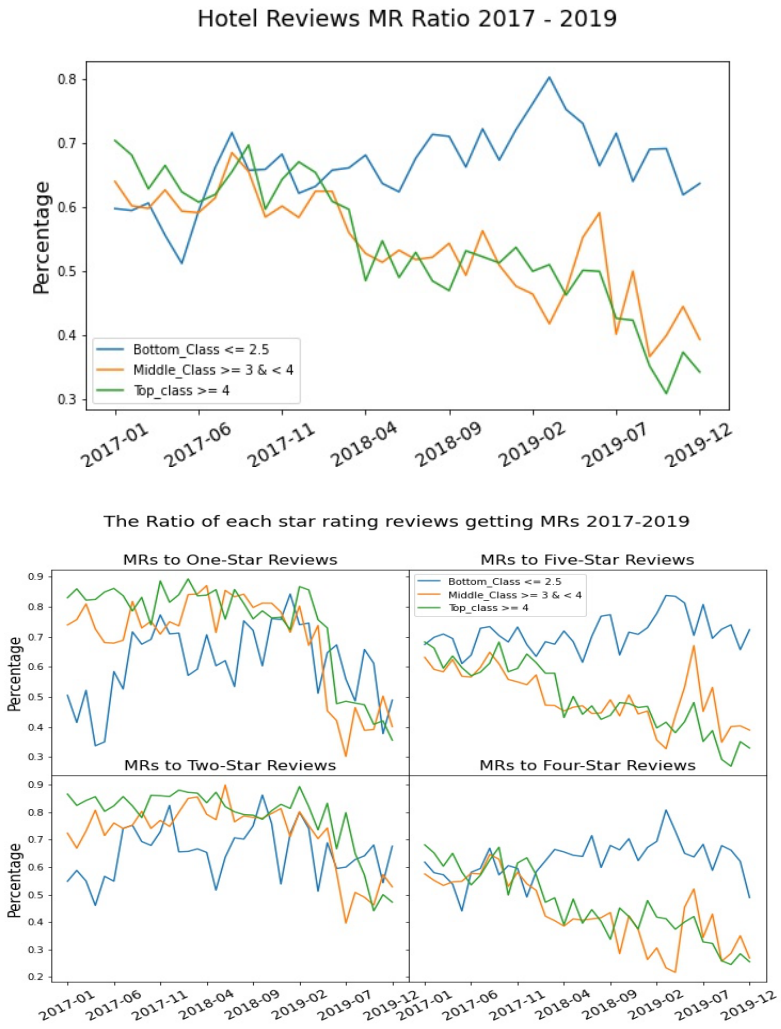


Figure 18 MRs ratio average over each month during 2017-2019



3.3 Variables

In this study, Table 15 presents the definitions and summary statistics of the variables. Table 16 and Table 17 show that the correlation among variables used in individual review dataset and hotel-monthly dataset, respectively, are mostly well below 0.8, indicating that the estimation is unlikely to be biased by multicollinearity issues. We also checked the VIF value of each variable before we estimated, making sure that VIFs are well under 10 and the estimation results are reliable.

Table 15 Variable definition and summary statistics

Variables	Definition	mean	std	min	median	max
HasMR	A dummy variable indicating whether a review receives a response from the hotel, with the value of 1 for yes and 0 for no. In this context, HasMR is the dependent variable in model 1.	0.530	0.500	0.000	1.000	1.000
HasMR*	At hotel-month panel data level, it means the monthly average percentage of how many reviews receive responses from the hotel. HasMR is an independent variable of interest in models 2 & 3.	0.580	0.440	0.000	0.795	1.000
MonthlyRating*	The average review star rating of a hotel in one month. The star rating of each review is the base, which is the overall review rating posted by a reviewer with 1 for "terrible", 2 for "poor", 3 for "average", 4 for "very good", and 5 for "excellent".	3.826	0.916	1.000	4.000	5.000
MonthlyRating_p1*	The average review star rating of a hotel in the last month.	3.832	0.906	1.000	4.000	5.000
MonthlyRating_GR*	The increase rate in monthly average review star rating of a hotel.	0.055	0.431	-0.800	0.000	4.000
Review Related						
Fake	A dummy variable indicating a review is identified as fake by a transferred deep learning algorithm, with the value of 1 for yes and 0 for no.	0.220	0.410	0.000	0.000	1.000
NFake	A dummy variable indicating a review is identified as non-fake by a transferred deep learning algorithm, with the value of 1 for yes and 0 for	0.410	0.490	0.000	0.000	1.000
Congruency	A dummy variable indicating whether the rating of a review and the monthly average rating of a hotel fall in the same category, positive or negative, with a value of 1 for yes and 0 for no.	0.660	0.470	0.000	1.000	1.000
Deviation	Difference between the rating of a review and the monthly average rating with a value of 0 for no difference and a positive value for the distance.	0.890	0.710	0.000	0.710	3.840
Managerial Response Related						
MR_WordCnt*	The number of words of a managerial response.	55.562	44.200	0.000	58.000	605.00
TextSimilarity*	Text Similarity between a consumer review and its corresponding managerial response with the value of 1 for identical and 0 for nothing in common.	0.039	0.040	0.000	0.034	0.545
Moderating Variable						

TopHotel	A dummy variable indicating whether the class of the hotel is 4.0 or above, with a value of 1 for yes and 0 for no.	0.610	0.490	0.000	1.000	1.000
Control Variables						
-Review Related						
HasPhoto	A dummy variable indicating whether a review has a photo, with a value of 1 for yes and 0 for no.	0.080	0.270	0.000	0.000	1.000
RevWordCnt	The number of words in a review.	111.250	111.930	9.000	70.000	3759
Rev_I_Cnt	The number of first-person pronouns there are in a review.	1.500	2.880	0.000	1.000	104.000
-Reviewer Related						
LnReviwerFriendsCnt	The logarithm of the number of followers a reviewer links to.	0.080	0.340	0.000	0.000	9.320
LnReviewerReviewsCnt	The logarithm of the number of reviews a reviewer has ever posted.	2.240	1.810	0.000	1.610	11.730
TripTypeBusiness	A dummy variable indicating whether the trip is for business, with the value of 1 for yes and 0 for no.	0.180	0.380	0.000	0.000	1.000
-Hotel Related						
LnMonthlyReviewsCount*	The logarithm of the number of reviews received by the focal hotel in	2.387	1.226	0.693	2.303	6.590

Notes:
* The asterisk indicates the variables are from panel dataset and are used for model 2&3 only. Other variables show their descriptive statistics at the review level. Most latter variables are used in both model 1 and models 2&3. When being used in models 2&3, the variables are aggregated to be the average values over each month.

Table 16 Correlation analysis of the variables in Model 1

	Variables	1	2	3	4	5	6	7	8	9	10	11	12
1	HasMR	1.000											
2	Fake	-0.046	1.000										
3	Nfake	0.044	-0.439	1.000									
4	Congruence	0.052	-0.009	0.009	1.000								
5	TopHotel	-0.028	0.005	0.006	0.054	1.000							
6	Deviation	0.110	0.026	-0.099	-0.302	-0.020	1.000						
7	HasPhoto	0.012	-0.155	0.323	0.003	0.023	-0.018	1.000					
8	RevWordCnt	0.069	-0.147	0.201	-0.019	0.010	0.115	0.159	1.000				
9	Rev_I_Cnt	0.059	-0.062	0.065	-0.024	0.014	0.163	0.083	0.666	1.000			
10	LnReviwerFriendsCnt	0.008	-0.110	0.214	0.001	0.016	-0.045	0.162	0.084	0.032	1.000		
11	LnReviewerReviewsCnt	0.040	-0.429	0.732	0.008	0.015	-0.134	0.265	0.245	0.082	0.398	1.000	
12	TripTypeBusiness	0.048	0.020	-0.022	0.037	0.089	0.024	-0.019	-0.031	0.078	0.018	-0.015	1.000

Table 17 Correlation analysis of the variables in Model 2 & 3

	Variables	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	MonthlyRating_GR	1.000														
2	Fake	0.002	1.000													
3	Nfake	0.024	-0.548	1.000												
4	HasMR	-0.048	0.033	-0.031	1.000											
5	MR_WordCnt	-0.110	0.016	-0.023	0.524	1.000										
6	TextSimilarity	-0.052	0.028	-0.024	0.462	0.601	1.000									
7	HasMR_p1	-0.019	0.030	-0.031	0.867	0.450	0.408	1.000								
8	MR_WordCnt_p1	0.030	0.027	-0.029	0.441	0.650	0.432	0.517	1.000							
9	TextSimilarity_p1	-0.044	0.015	-0.014	0.400	0.436	0.510	0.458	0.599	1.000						
10	MonthlyRating_p1	-0.543	-0.014	0.028	0.152	0.181	0.220	0.135	0.090	0.203	1.000					
11	HasPhoto	-0.016	-0.201	0.319	-0.026	-0.016	-0.011	-0.025	-0.018	-0.006	-0.016	1.000				
12	RevWordCnt	-0.059	-0.185	0.214	-0.038	0.029	0.005	-0.039	-0.035	-0.034	-0.096	0.199	1.000			
13	Rev_I_Cnt	-0.060	-0.072	0.065	0.000	0.038	0.022	0.000	-0.005	-0.020	-0.105	0.115	0.654	1.000		
14	TripTypeBusiness	-0.055	0.055	-0.043	0.124	0.117	0.117	0.124	0.116	0.142	0.099	0.011	-0.048	0.092	1.000	
15	LnMonthlyReviewsCount	-0.124	0.152	-0.189	0.071	0.241	0.192	0.071	0.236	0.184	0.287	-0.085	-0.138	-0.072	0.072	1.000

3.3.1 Fake/NFake review identification

Based on the results of the first essay of this dissertation, we transfer and use a pre-trained fake review classifier to label the sample data in this study. Using a deep learning algorithm, MLP (Multi-Layer Perceptron) and outperforming several other detectors, the classifier was pre-trained on a labeled large dataset from Yelp.com, getting 85% accuracy. The attributes used as inputs of the model include 20,000 top occurred bigram (a 128-dimension vector represents each bigram) and other 112 numerical features related to review text, reviewer profile, and hotel characteristics. We carried out the labeling process on Google Colab Pro. To make the results more convincing, we first reran the model with the original Yelp data with five different randomly shuffling operations, then got five very similar classifiers but having some difference in the coefficient estimations of variables (features), after that applied all the five classifiers to the review dataset used in this study, got predicted probability of each review being fake, last extracted the intersection amongst the predictive results from five classifiers as the labels assigned to each review. Since the five predicted values for each review by applying the above five classifiers are not identical to each other, thus we have a third label other than Fake and non-fake, which we named as undecided.

Table 18 shows the results of fake and non-fake identification for each review rating level. We categorized 22% of the sample reviews as fake, 41% as non-fake, and the remaining 37% as undecided. In contrast to the 3- and 4-star reviews, extreme reviews have a relatively higher percentage to be labeled as fake (25% for 5-star and 23% for 1-star), and lower as non-fake (37% for 5-star and 28% for 1-star). The above distribution is in line with the previous studies (Mukherjee et al., 2011.) , Furthermore, we also noticed from Figure 19 and Figure 20 that there was a time trend in the percentage of fake and non-fake reviews for hotels from

different hotel class categories. Overall, hotels of middle and above class, the portion of fake had been increasing and non-fake decreasing, while the bottom-class hotels displayed the opposite trend. For instance, in January 2017 only 15% and 17% of extreme 1-star and 5-star reviews from top-class hotels were identified as fake, but in December 2019 the percentage increased to 24% and 32%, while the non-fake percentage decreased from 47% to 23% (1-star) and from 49% to 28% (5-star). In contrast, the bottom-class hotels presented a relatively stable and even opposite trend. Detailed monthly average information can be seen from Table 33 and Table 34 in the Appendix.

Table 18 Sample description based on Fake/NFake identification

Review Rating	Number of Reviews	Fake	NFake	Fake Ratio	NFake Ratio
1	25,435	5,901	7,136	23%	28%
2	23,406	4,565	8,708	20%	37%
3	43,736	7,747	20,279	18%	46%
4	83,956	14,793	42,741	18%	51%
5	170,084	42,365	63,184	25%	37%
Total	346,617	75,371	142,048	22%	41%

Figure 19 Fake/NFake ratio comparison in Jan 2017 vs Dec 2019

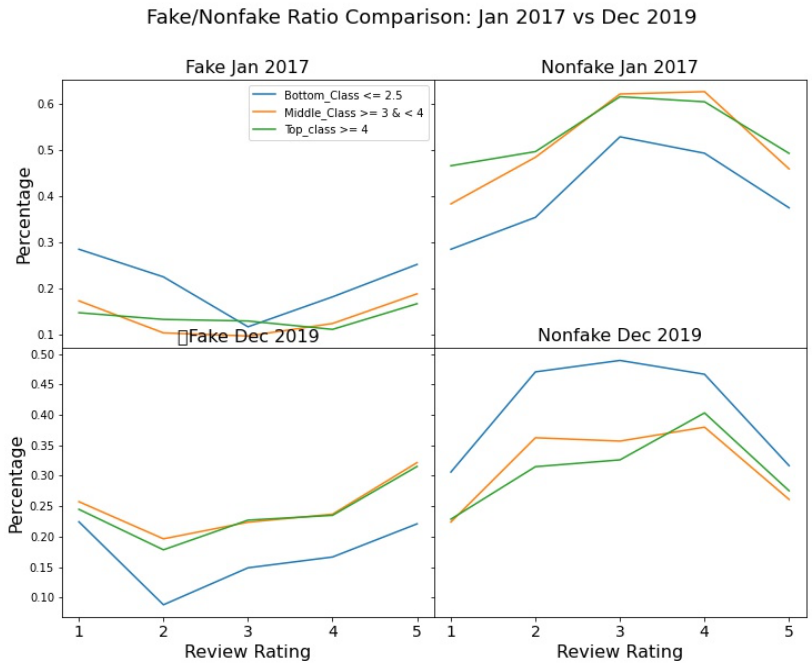
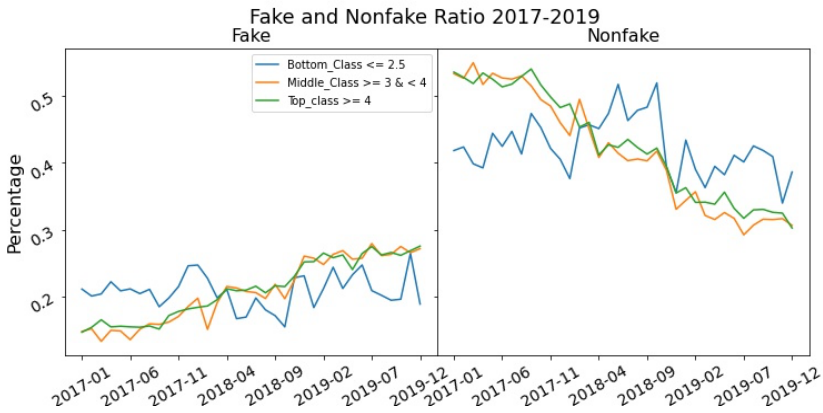


Figure 20 Fake/NFake Ratio 2017-2019



3.3.2 Text similarity between review and response

There are multiple widely recognized functions to calculate textual similarity, from Cosine similarity, Euclidean distance, to Jarccard similarity, and so on. In this study, we adopted Cosine similarity as the method, and used TF-IDF score (term frequency-inverse document frequency) as the value of each entry(word) in the vector representing review or response text. The study by Z. Zhang et al.(2019), takes 1 as the value of the entry in the vector for the words that are present in the document when examining the Cosine similarity among managerial responses. The reason for not using Word2vec methods, in which a word is represented as a multidimensional array rather than a single scaler number like in TF-IDF score, is that the vectors representing words are trained out of a more general corpus rather than a pinpointed corpus about hotel online reviews and responses.

Cosine similarity uses the cosine of the angle to measure the textual similarity between two non-zero vectors of an inner product space. The higher the cosine, the closer two vectors, representing the higher similarity between vectors.

Technically, we first tokenized every review and response, removed stop words, used the top frequent 20,000 unique unigrams and bigrams which appear in the reviews as the

vocabulary, then vectorized by using a bag of words with TF-IDF scores to convert each text (review and response) into a vector, lastly calculated the cosine similarity between each pair of review and response. Through the process, we adopted embedded methods and functions such as TfidfVectorizer and pairwise.cosine_similarity, from Scikit-learn, which is a free software machine learning library for Python.

The function for calculating the text similarity between review and response is as follows:

$$Cosine\ Similarity = \frac{Rev \cdot Res}{\| Rev \| \| Res \|} = \frac{\sum_{i=1}^{n} Rev_i Res_i}{\sqrt{\sum_{i=1}^{n} {Rev_i}^2}\ \sqrt{\sum_{i=1}^{n} {Res_i}^2}} \quad (1)$$

Where n means the size of the vocabulary. In this study, we assigned n = 20,000. We then took Cosine similarity score to measure the text similarity between one review and its corresponding response.

Table 19 is an example to illustrate the above process. For simplicity but without losing generalization, we use term frequency to represent each word in the document:

The review text says: “good hotel, great service, good experience.”

The response text says: “great to see you had a great experience.”

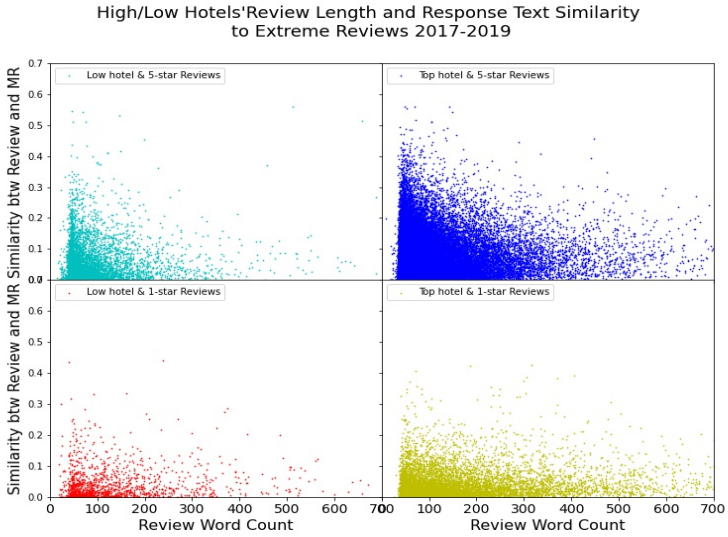
The first two rows are the vectors representing the review and response, with term frequency as the value of each word (feature). Then we calculate the L2 norms for each vector. After that, we get the normalized vector by dividing each entry by L2 norms, shown in the third and fourth rows of Table 19. Last, we calculate the cosine similarity with a dot product, which is .335. In this study, the text similarity score used in this study is calculated in the same way, the only difference is what we used to represent the words is their TF-IDF scores rather than term frequencies.

Table 19 An example of Cosine similarity between a pair of review and response

Vocabulary	a	experience	good	great	had	hotel	see	service	to	you	L2 norms
Review vector	0	1	2	1	0	1	0	1	0	0	2.828
Response vector	1	1	0	2	1	0	1	0	1	1	3.162
Normalized review vector	0.000	0.354	0.707	0.354	0.000	0.354	0.000	0.354	0.000	0.000	
Normalized response vector	0.316	0.316	0.000	0.632	0.316	0.000	0.316	0.000	0.316	0.316	**Cosine**
Dot product	0.000	0.112	0.000	0.224	0.000	0.000	0.000	0.000	0.000	0.000	0.335

Figure 21 shows the distribution of text similarity between review and response of extreme positive rating, 5-star, and extreme negative rating, 1-star. It seems that a shorter or a positive review or a review about a top-class hotel correlates with a relatively higher text similarity.

Figure 21 Text similarity vs review length



3.4 Empirical models

We employed a logistic regression to estimate Model 1 in this study and tested the research hypotheses of H1 when using a binary variable if a review receives a response from the hotel as the dependent variable. When the test goes to the impact of MRs on the overall monthly review rating level or its growth rate, we used the OLS regressions to estimate Models 2 & 3, in which hotel fixed effects were included to control hotel heterogeneity.

We specify the econometric models as follows:

$$HasMR_i = \beta_{10} + \beta_{11}Fake_i + \beta_{12}NFake_i + \beta_{13}Deviation_i + \beta_{14}Congruence_i + \beta_{15}TopHotel_i + \beta_{16}Congruence_i * TopHotel_i + \boldsymbol{\gamma_1}Controls_i + \varepsilon_{1i}$$

(Model 1)

$$MonthlyRating_{jt} = \beta_{20} + \beta_{21}Fake_{jt} + \beta_{22}NFake_{jt} + \beta_{23}HasMR_{jt} + \beta_{24}MRWordCnt_{jt} + \beta_{25}TextSimilarity_{jt} + \boldsymbol{\gamma_2}Controls_{jt} + \mu_j + \varepsilon_{2jt}$$

(Model 2)

$$MonthlyRatingGR_{jt} = \beta_{30} + \beta_{31}Fake_{jt} + \beta_{32}NFake_{jt} + \beta_{33}HasMR_{jt} + \beta_{34}MRWordCnt_{jt} + \beta_{35}TextSimilarity_{jt} + \alpha_{31}MonthlyRating_{jt-1} + \boldsymbol{\gamma_3}Controls_{jt} + \mu_j + \varepsilon_{3jt}$$

(Model 3)

The definition and descriptive statistics of each variable are shown in Table 15. In Model 1, our unit of analysis is each review and response. The dependent variable is binary: a review got a response or not; the independent variables include Fake, NFake, Congruency, and Deviation; the moderating variable is TopHotel, a dummy variable indicating if the hotel is top-class, and thus the interaction term, Congruency * TopHotel is also included in the model. The control variables in Model 1 are $HasPhoto$, $RevWordCnt$, $RevICnt$, $LnReviewerFriendsCnt$, $LnReviewerReviewsCnt$, and $TripTypeBusines$. Our objective is to estimate the significance level of $\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, and\ \beta_{16}$, among which the first four represent the direct effects of each independent variable on the odds for a review to receive an MR, while β_{16} reflects the moderating effect of the hotel class. ε_{1i} represents the error term.

In Models 2 & 3, the unit of our analysis is hotel-month, based on a hotel-month panel dataset. We used fixed-effects panel data model proposition to decode the interplay of fakeness identification, managerial response, and hotel review rating and its growth. The dependent variable in Model 2 is $MonthlyRating_{jt}$, which is the average review rating across all consumer reviews for hotel j in month t, and the dependent variable in Model 3 is its growth rate,

$MonthlyRatingGR_{jt}$, which measures the increase in the average monthly review rating for hotel j in month t. The control variables in Models 2 & 3 are the monthly average of $HasPhoto$, $RevWordCnt$, $RevICnt$, and $TripTypeBusiness$, and $LnMonthlyReviewCont$ for month t. The independent variables in Models 2 & 3 are the same: $Fake_{jt}$, $NFake_{jt}$, $HasMR_{jt}$, $MRWordCnt_{jt}$, and $TextSimilarity_{jt}$, which are the monthly average of each original variables for hotel j in month t. For instance, $HasMR_{jt}$ measures the managerial response ratio for hotel j in month t, and others are similarly measure the average level of each variable across hotel and month. Our objective is to estimate the significance level of $\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}, \beta_{25}$ to see the associations between the level of each independent variable and hotel rating and its growth rate. $MonthlyRating_{jt-1}$ is a lagged variable representing the average monthly rating for hotel j in month $t-1$; $Hotel_j$ is the unobservable hotel-specific effects hotel j; ε_{2jt} denotes the error term varying across hotel and time.

Prior studies indicated that earlier reviews and MRs usually influence consumer behavior, the effects of word of mouth often carry over. For instance, Xie et al. (2016) used lagged one-quarter effect of MRs on examining the hotel's reputation and performance, and Zhang et al. (2020) also used a month lag effect of MRs on examining the influence of managerial responses to reviews on online hotel booking volume. This study did not use the lag effect for granted. Instead, we first examined whether the formerly used one month lag effect exists or not, using the first-order lag of each managerial response variable only. In other word, plug $HasMR_{jt-1}$, $MRWordCnt_{jt-1}$, and $TextSimilarity_{jt-1}$ into Model 2 instead of $HasMR_{jt}$, $MRWordCnt_{jt}$, and $TextSimilarity_{jt}$ to see the effects. Table 20 shows that the overall goodness-of-fit of the lag effect only. The first column shows when only Fake/NFake ratio and one month lag MR variables are included, the explanatory power is significant but very small by reading R-Squared (.0042) and F-statistic (10.347, p = 0.000). The first column also shows that $HasMR_{jt-1}$

presents significant positive association with hotel rating in month t (Coefficient = .0078, p < 0.01). After adding the control variables, as shown in the second column of Table 20, the R-Squared of the model increased by ten times (.0518), and the coefficient of $HasMR_{jt-1}$ stays significantly positive.

Table 20 Pretest of Model 2 (MR lagged variables only)

Dep. Variable	**MonthlyRating**	
	(a)	
	Model 2.pre	**Model 2.pre1**
Estimator	PanelOLS	PanelOLS
No. Observations	12624	12624
Cov. Est.	Robust	Robust
R-squared	0.0042	0.0518
R-Squared (Within)	0.0042	0.0518
R-Squared (Between)	0.0046	0.1326
R-Squared (Overall)	0.0117	0.0940
F-statistic	10.347	66.407
P-value (F-stat)	0.0000	0.0000
==========	==========	==========
Independent Variables:		
Fake	0.1638**	0.1157*
	(0.0639)	(0.0626)
NFake	0.1482***	0.2058***
	(0.0401)	(0.0408)
Lagged MR Variables:		
HasMR_p1	0.0778***	0.0856***
	(0.0213)	(0.0209)
MR_WordCnt_p1	4.591e-05	2.336e-05
	(0.0002)	(0.0002)
TextSimilarity_p1	-0.1207	-0.0929
	(0.2226)	(0.2196)
Control Variables:		
HasPhoto		-0.0699
		(0.057)
RevWordCnt		-0.0015***
		(0.000)
Rev_I_Cnt		-0.0418***
		(0.008)
TripTypeBusiness		-0.1164**
		(0.048)
LnMonthlyReviewsCount		0.0530***
		(0.013)
Const	3.8299***	3.8284***
	(0.0056)	(0.0055)
==========	==========	==========
Effects	Entity	Entity

Note: Coefficients are shown in the table; standard errors are reported in parentheses.
*p<0.10; **p<0.05; ***p<0.01

A recent study examines the relationship between hotel rating and the response percentage to critical reviews in the same time window (Zhu et al., 2021). In this study, based

on the above results, which show the model only carries a very small explanatory power when using the lagged variable only, we then tested the contemporaneous effects to find a specification to improve the explanatory power of Model 2. There are other considerations behind it. First, most responses (87%) were posted within the same month, which means most responses to other reviews, which were posted in the same month, are visible to the subsequent reviewers before they posted their own. Second, in contrast to old news, people tend to pay more attention to recent information. So, we then replaced the lagged variables with the unlagged variables, as shown in Table 21.

As we expected, it shows that the overall goodness-of-fit enhanced greatly: R-Squared increased from .0042 to .0397. The first column of Table 21 shows the detail when only the Fake/NFake ratio and MR variables of the current month are included. The first column also shows that, in contrast to the one-month lag effect (Coefficient = .0778, $p < 0.01$) $HasMR_{jt}$ presents a much larger positive association with hotel rating in month t (Coefficient = .1257, $p < 0.01$). When control variables are included, as shown in the second column of Table 21, the R-Squared of the model increased even further (.0781), and the coefficient of $HasMR_{jt-1}$ stays as significantly positive and larger.

Table 21 Pretest of Model 2 (Contemporaneous MR variables only

Dep. Variable	**MonthlyRating**	
	(a)	
	Model 2.1	**Model 2.2**
Estimator	PanelOLS	PanelOLS
No. Observations	13081	13081
Cov. Est.	Robust	Robust
R-squared	0.0397	0.0781
R-Squared (Within)	0.0397	0.0781
R-Squared (Between)	-0.0941	0.0609
R-Squared (Overall)	-0.0206	0.0742
F-statistic	104.44	106.92
P-value (F-stat)	0.0000	0.0000
==========	==========	==========
Independent Variables:		
Fake	0.1628***	0.1174*
	(0.0612)	(0.0602)
NFake	0.1395***	0.1932***
	(0.0381)	(0.0388)

MR Variables:		
HasMR	0.1257***	0.1206***
	(0.0223)	(0.0217)
MR_WordCnt	-0.0045***	-0.0042***
	(0.0003)	(0.0003)
TextSimilarity	1.5737***	1.6718***
	(0.2531)	(0.2615)
Control Variables:		
HasPhoto		-0.0449
		(0.055)
RevWordCnt		-0.0012***
		(0.000)
Rev_I_Cnt		-0.0376***
		(0.008)
TripTypeBusiness		-0.1151**
		(0.046)
LnMonthlyReviewsCount		0.0665***
		(0.013)
Const	3.8259***	3.8259***
	(0.0054)	(0.0053)
==========	==========	==========
Effects	Entity	Entity

Note: Coefficients are shown in the table; standard errors are reported in parentheses.

*p<0.10; **p<0.05; ***p<0.01

Based on the above and to accommodate both the current and the lag effects, we include both in the model. In this study, the specifications of Models 2 & 3 would be, both $HasMR_{jt-1}$, $MRWordCnt_{jt-1}, and\ TextSimilarity_{jt-1}$ and $HasMR_{jt}$, $MRWordCnt_{jt}, and\ TextSimilarity_{jt}$ are simultaneously plugged in to see the combined effects. Therefore, it is very important to note that when evaluate the effect of these three variables, $HasMR,\ MRWordCnt,\ and\ TextSimilarit$, we need to see them in a collective way, especially when negative current effect and positive lag effect happen simultaneously or vice versa, we need to further check the magnitude of the coefficient and then conclude a collective negative or positive effect the variable carries.

Before estimating to make sure all the variables are stationary and no unit root is in there, we employed both unit root tests: Augmented Dickey-Fuller test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. The results of both tests showed that all variables in Models 2 & 3 are stationary, ruling out the possibility of spurious regression.

4 Findings and results

In Model 1, we model the chance of getting a response at review level as a function of Fake/NFake detection, congruency, deviation, and the interaction between hotel class and the congruency of review. In Model 2, we model the hotel rating, and its growth at a hotel-month level as a function of Fake/NFake detection and major attributes of managerial responses. When estimating, we control for multiple relevant attributes of review-, reviewer-, and hotel- related. Furthermore, in every specification of Models 2 & 3we carried out and past Durbin-Watson test and calculated VIF for each variable, ruling out the possible issues due to autocorrelation and multicollinearity. In addition, after each estimation is done, we carried out a Hausman test to determine (Hausman, 1978) and confirm our choice on fixed effects specification of the models. The results of Hausman test rejected the original hypothesis for both Models 2 & 3. Data analysis was conducted in Jupyter Notebook with Python.

4.1 Results of Model 1

4.1.1 Statistic description and correlation analysis

We presented in subsection 3.3 the detailed statistic description of variables in Model 1. Table 15 shows on average 53% reviews received managerial response, and Table 12, Table 13, and Table 14 show the response ratio varies across time, hotel class, and review rating. Table 16 shows the details of variable correlations.

4.1.2 Estimation results and hypothesis testing

Table 22 displays the estimation results of Model 1 with the dependent variable as to whether a review receives an MR from the hotel. In Table 22, Model 1.0 includes only independent variables, Model 1.1 contains independent and control variables, Model 1.2 includes

independent and moderating variables, and Model 1.3 contains all variables. The coefficients' signs for all variables of interest remain the same across the four columns of Table 22.

The estimation results show that the coefficient of Fake is significantly negative (coefficient = - .116 < 0, p < .001), indicating that being identified as a fake review is less likely to receive an MR. Therefore, H1a is supported. The coefficient of NFake is significantly positive (coefficient = .105 > 0, p < .001), indicating that being labeled as non-fake a review is more likely to receive an MR. So, H1b holds.

The estimation results also show that the coefficient of Congruence is significantly positive (coefficient = .488>0, p < .001), indicating that being in the same positive or negative rating category as the overall hotel review rating a review is more likely to receive an MR. Therefore, H1c is supported. We also found the same significant positive effect on Deviation (coefficient = .405>0, p < .001), meaning that a review with a larger deviation is more likely to receive an MR, so, H1d is supported.

With regard to the moderating role of hotel class, the estimation results show that positive association between congruence and the probability of receiving an MR is attenuated when the review is about a top-class hotel (coefficient = -.130 < 0, p < .001). Therefore, H1e is supported.

Table 22 Estimation results of Model 1

Model_1 Main Results

Dependent variable: HasMR

	Model_1.0	Model_1.1	Model_1.2	Model_1.3
Independent Variables:				
Fake	-0.146***	-0.119***	-0.143***	-0.116***
	(0.009)	(0.009)	(0.009)	(0.009)
NFake	0.186***	0.108***	0.186***	0.105***
	(0.008)	(0.011)	(0.008)	(0.011)
Congruence	0.414***	0.404***	0.511***	0.488***
	(0.008)	(0.008)	(0.012)	(0.012)
Deviation	0.426***	0.409***	0.422***	0.405***

	(0.005)	(0.006)	(0.005)	(0.006)
Moderating Variables:				
Congruence * TopHotel			-0.152***	-0.130***
			(0.015)	(0.015)
TopHotel			-0.029*	-0.063***
			(0.012)	(0.012)
Control Variables:				
HasPhoto		-0.069***		-0.060***
		0.014)		(0.014)
RevWordCnt		0.001***		0.001***
		0.000		0.000
Rev_I_Cnt		0.005**		0.005**
		(0.002)		(0.002)
LnReviewerFriendsCnt		-0.039***		-0.038***
		(0.011)		(0.011)
LnReviewerReviewsCnt		0.026***		0.027***
		(0.003)		(0.003)
TripTypeBusiness		0.236***		0.252***
		(0.009)		(0.009)
Const	0.136***	0.137***	0.136***	0.138***
	(0.003)	(0.003)	(0.003)	(0.003)
Observations	346,617	346,617	346,617	346,617
Pseudo R-squared:	0.018	0.021	0.019	0.022
Residual Std. Error	1.000 (df=346612)	1.000 (df=346607)	1.000 (df=346610)	1.000 (df=346604)
LLR p-value:	0.000	0.000	0.000	0.000

Notes: Coefficients are shown in the table; standard errors are reported in parentheses.
*p<0.05; **p<0.01; ***p<0.001

Table 23 shows the average marginal effects of Model 1.3 in Table 23, which measures the impact that an instantaneous unit change in one variable has on the outcome variable (average derivatives) while all other variables are as observed. For a review which is labeled as fake, it will be 2.81% less likely to receive an MR, while being non-fake or congruent, the chance of getting an MR increases by 2.55% or 11.8%, respectively. Also, one more point deviation from the monthly average review rating, a review will be 9.8% more likely to receive an MR. If a review which is about a top-class hotel, the chance will decrease by 1.52%, and if the rating of the review is congruent with that of the top hotel, the chance will further decrease by 3.14%.

Table 23 Marginal effects of Model 1

Logit Marginal Effects				Dep. Variable: HasMR		
	dy/dx	std err	z	P>|z|	[0.025	0.975]

Independent Variables:						
Fake	-0.0281***	0.002	-12.269	0.000	-0.033	-0.024
NFake	0.0255***	0.003	9.744	0.000	0.020	0.031
Congruency	0.1180***	0.003	42.160	0.000	0.112	0.123
Deviation	0.0980***	0.001	75.510	0.000	0.095	0.101
Moderating Variables:						
TopHotel	-0.0152***	0.003	-5.211	0.000	-0.021	-0.009
Congruency*TopHotel	-0.0314***	0.004	-8.702	0.000	-0.039	-0.024
Control Variables:						
HasPhoto	-0.0145***	0.003	-4.437	0.000	-0.021	-0.008
RevWordCnt	0.0002***	1.09e-05	17.168	0.000	0.000	0.000
Rev_I_Cnt	0.0012**	0.000	2.779	0.005	0.000	0.002
LnReviewerFriendsCnt	-0.0091***	0.003	-3.318	0.001	-0.014	-0.004
LnReviewerReviewsCnt	0.0065***	0.001	8.694	0.000	0.005	0.008
TripTypeBusiness	0.0609***	0.002	27.362	0.000	0.057	0.065

Notes: **p<0.01; ***p<0.001

4.1.3 Robustness check

We employed the Probit model to check the findings of Model 1 as extracted from above. We showed the estimation results and marginal effects in Table 22 and Table 23. Comparing Table 24 to Table 22, and Table 25 and Table 23 are quantitatively very close, suggesting that our models are robust.

Table 24 Estimation results of Model 1 (Probit)

Model_1 Robustness Check Dependent variable: HasMR				
	Model_1	**Model_1.1**	**Model_1.2**	**Model_1.3**
Independent Variables:				
Fake	-0.091***	-0.074***	-0.089***	-0.072***
	(0.006)	(0.006)	(0.006)	(0.006)
NFake	0.116***	0.067***	0.116***	0.066***
	(0.005)	(0.007)	(0.005)	(0.007)
Congruence	0.257***	0.250***	0.317***	0.302***
	(0.005)	(0.005)	(0.007)	(0.007)
Deviation	0.264***	0.254***	0.262***	0.251***
	(0.003)	(0.003)	(0.003)	(0.003)
Moderating variables:				
Congruence * TopHotel			-0.095***	-0.081***
			(0.009)	(0.009)
TopHotel			-0.017*	-0.038***
			(0.007)	(0.008)
Control Variables:				
HasPhoto		-0.042***		-0.037***
		(0.008)		(0.008)
RevWordCnt		0.001***		0.000***

		0.000		0.000
Rev_I_Cnt				0.002*
				(0.001)
LnReviewerFriendsCnt		-0.025***		-0.024***
		(0.007)		(0.007)
LnReviewerReviewsCnt		0.017***		0.017***
		(0.002)		(0.002)
TripTypeBusiness		0.149***		0.157***
		(0.006)		(0.006)
Const	0.084***	0.085***	0.085***	0.085***
	(0.002)	(0.002)	(0.002)	(0.002)
Observations	346,625	346,625	346,625	346,625
Pseudo R-squared:	0.018	0.021	0.019	0.022
Residual Std. Error	1.000 (df=346620)	1.000 (df=346614)	1.000 (df=346618)	1.000 (df=346612)
LLR p-value:	0.000	0.000	0.000	0.000

Notes: Coefficients are shown in the table; standard errors are reported in parentheses. *p<0.05; **p<0.01; ***p<0.001

Table 25 Marginal effects of Model 1 (Probit)

Probit Marginal Effects				Dep. Variable: HasMR		
	dy/dx	std err	z	P>|z|	[0.025	0.975]
----------	---	---	---	---	---	---
Independent Variables:						
Fake	-0.0281***	0.002	-12.261	0	-0.033	-0.024
NFake	0.0255***	0.003	9.743	0	0.020	0.031
Congruency	0.1174***	0.003	42.012	0	0.112	0.123
Deviation	0.0975***	0.001	76.150	0	0.095	0.100
Moderating Variables:						
TopHotel	-0.0149***	0.003	-5.102	0	-0.021	-0.009
Congruency*TopHotel	-0.0316***	0.004	-8.736	0	-0.039	-0.024
Control Variables:						
HasPhoto	-0.0144***	0.003	-4.399	0	-0.021	-0.008
RevWordCnt	0.0002***	1.06E-05	17.014	0	0.000	0.000
Rev_I_Cnt	0.0010**	0.000	2.358	0.018	0.000	0.002
LnReviewerFriendsCnt	-0.0093***	0.003	-3.384	0.001	-0.015	-0.004
LnReviewerReviewsCnt	0.0067***	0.001	8.893	0	0.005	0.008
TripTypeBusiness	0.0609***	0.002	27.420	0	0.057	0.065
Notes: **p<0.01; ***p<0.001						

4.2 Results of Models 2 & 3

4.2.1 Statistic description and correlation analysis

Statistic description of each variable used in Models 2 & 3 can be found in subsection 3.3. Table 17 shows the details of variable correlations.

4.2.2 Estimation results

4.2.2.1 Model 2

We estimated Model 2 on both the entire hotel-month panel dataset to obtain main results and four meaningful segments of the data to check the robustness of our estimates.

Table 26 displays the estimation results of Model 2, whose dependent variable is the MonthlyRating, the average monthly hotel rating. The first column of Table 26 includes only independent variables and lagged variables, and the second column contains all variables. The second column, Model 2.4, shows that the coefficient of Fake is marginally significantly positive (coefficient = .1050, $p < 0.10$), and the coefficient of NFake is significantly positive (coefficient = .2053, $p < 0.01$) when we included the control variables.

Thus, both the percentage of Fake and NFake reviews a hotel receives in a month significantly and positively impact hotel rating, suggesting that Hypothesis H2b holds, but H2a does not. In an extreme situation, when a hotel changes its percentage of non-fake review from zero to 100%, ceteris paribus, its rating will increase by .21 point.

With regard to the MR related independent variables, Table 26 shows that HasMR has a positive effect at one lag (coefficient = .1098, $p < .01$) and no significant contemporaneous effect, MRWordCnt has a much larger negative contemporaneous effect (coefficient = -.0044, $p < .01$) and a smaller positive effect at one lag (coefficient = .0010, $p < .01$) , and TextSimilarity has a large positive contemporaneous effect (coefficient = 1.6317, $p < .01$) but no significant lag effect. In other words, a higher level of MR ratio, shorter MRs, and higher text similarity between MRs and reviews, associate with a higher hotel rating. Therefore, the Hypotheses H2c, H2d, and H2e hold. The magnitude of positive coefficient of TextSimilarity is big. Imagine increasing text similarity between review and response from the average value .039 up by .10 to .139, ceteris paribus, the rating will go upward by .16.

For the control variables, the coefficients of RevWordCnt, Rev_I_Cnt, and TripTypeBusiness are negative with significance, while the volume of monthly review a hotel

has a significantly positive coefficient, but the ratio of photo review show no significant effect. Specifically, when there are a larger portion of lengthy reviews, reviews using more first-person pronouns, and reviews about business trips, a hotel is likely to get a lower monthly rating, in contrast to a larger volume of reviews associates with a higher monthly rating.

Table 26 Estimation results of Model 2

Dep. Variable	**MonthlyRating**	
	(a)	
	Model 2.3	**Model 2.4**
Estimator	PanelOLS	PanelOLS
No. Observations	12624	12624
Cov. Est.	Robust	Robust
R-squared	0.0460	0.0868
R-Squared (Within)	0.0460	0.0868
R-Squared (Between)	-0.0572	0.0974
R-Squared (Overall)	-0.0006	0.0926
F-statistic	73.258	88.880
P-value (F-stat)	0.0000	0.0000
==========	==========	==========
Independent Variables:		
Fake	0.1494**	0.1050*
	(0.0626)	(0.0614)
NFake	0.1496***	0.2053***
	(0.0390)	(0.0397)
MRs Variables:		
HasMR	0.0353	0.0230
	(0.0325)	(0.0314)
MR_WordCnt	-0.0048***	-0.0044***
	(0.0003)	(0.0003)
TextSimilarity	1.5315***	1.6317***
	(0.2627)	(0.2730)
Lag MRs Variables:		
HasMR_p1	0.0999***	0.1098***
	(0.0308)	(0.0299)
MR_WordCnt_p1	0.0012***	0.0010***
	(0.0002)	(0.0002)
TextSimilarity_p1	-0.1155	-0.1147
	(0.2214)	(0.2188)
Control Variables:		
HasPhoto		-0.0722
		(0.055)
RevWordCnt		-0.0013***
		(0.000)
Rev_I_Cnt		-0.0407***
		(0.008)
TripTypeBusiness		-0.1087**
		(0.047)
LnMonthlyReviewsCount		0.0647***
		(0.013)
Const	3.8304***	3.8288***
	(0.0055)	(0.0054)
==========	==========	==========

Effects	Entity	Entity

Note: Coefficients are shown in the table;
*p<0.10; **p<0.05; ***p<0.01

4.2.2.2 Model 3

Like what we did for Model 2, we estimated Model 3 on both the entire hotel-month panel dataset to obtain main results and four meaningful segments of the data to serve as a robustness check.

Table 27 displays the estimation results of Model 3, whose dependent variable is the MonthlyRatingGR, the growth rate of monthly hotel rating. The first column of Table 27 includes only independent variables and lagged variables, and the second column contains all variables. The second column, Model 3.4, shows that the coefficient of Fake is not significant, and the coefficient of NFake is significantly positive (coefficient = .0845, $p < 0.01$) when we included the control variables. Thus, the percentage of NFake reviews a hotel receives in a month has a significant and positive impact on hotel rating growth rate, suggesting that Hypothesis H3b is held but H3a is not. In an extreme situation, when a hotel's percentage of non-fake review increases from zero to 100%, ceteris paribus, its rating growth rate can be upward huge, by 8.45%, considering the average growth rate is only 5.5%.

In terms of the MR related independent variables, the findings are very similar to that of Model 2. Table 27 shows that HasMR has a positive effect at one lag (coefficient = .0440, $p < .01$) but no contemporaneous effect, MRWordCnt has a negative contemporaneous effect at one lag (coefficient = -.0014, $p < .01$) but no contemporaneous effect, and TextSimilarity has a large positive contemporaneous effect (coefficient = .7696, $p < .01$ but no significant lag effect. That is to say, higher level of MR ratio, shorter MRs, and higher text similarity between MRs and reviews, associate with a higher hotel rating growth rate. Therefore, the Hypotheses H3c, H3d, and H3e hold still. The magnitude of positive association of TextSimilarity is exciting.

Considering increasing text similarity between review and response from the average value .039 up by .10 to .139, ceteris paribus, the rating increase rate will go upward by a huge 7.696%, considering the average growth rate is only 5.5%.

For the control variables, the findings are also highly similar to the results of Model 2. The coefficients of RevWordCnt, Rev_I_Cnt, and TripTypeBusiness are negative with significance, while the volume of monthly review and the ratio of photo review show no significant impact. Specifically, when there are a larger portion of lengthy reviews, reviews using more first-person pronouns, and reviews from business trip, a hotel is likely to have a lower monthly rating.

Table 27 Estimation results of Model 3

Dep. Variable	MonthlyRatingGR	MonthlyRatingGR
Estimator	PanelOLS	PanelOLS
No. Observations	12624	12624
Cov. Est.	Robust	Robust
R-squared	0.4471	0.4591
R-Squared (Within)	0.4471	0.4591
R-Squared (Between)	-2.0815	-1.8724
R-Squared (Overall)	0.1351	0.1687
F-statistic	1092.5	736.80
P-value (F-stat)	0.0000	0.0000
Independent Variables:		
Fake	0.0335	0.0182
	(0.0334)	(0.0332)
NFake	0.0640***	0.0845***
	(0.0215)	(0.0219)
MRs Variables:		
HasMR	0.0201	0.0139
	(0.0170)	(0.0168)
MR_WordCnt	-0.0016***	-0.0014***
	(0.0001)	(0.0001)
TextSimilarity	0.7302***	0.7696***
	(0.1355)	(0.1386)
Lag MRs Variables:		
HasMR_p1	0.0407**	0.0440***
	(0.0172)	(0.0171)
MR_WordCnt_p1	0.0002	0.0002
	(0.0001)	(0.0001)
TextSimilarity_p1	-0.1375	-0.1333
	(0.1104)	(0.1108)
Lag Rating Variable:		
MonthlyRating_p1	-0.4510***	-0.4519***
	(0.0114)	(0.0112)

Control Variables:		
HasPhoto		-0.0337
		(0.030)
RevWordCnt		-0.0005***
		(0.000)
Rev_I_Cnt		-0.0138***
		(0.005)
TripTypeBusiness		-0.0489**
		(0.023)
LnMonthlyReviewsCount		0.0095
		(0.007)
Const	0.0551***	0.0546***
	(0.0028)	(0.0028)
Note: Coefficients are shown in the table; standard errors are reported in parentheses.		
*p<0.10; **p<0.05; ***p<0.01		

4.2.3 Robustness check

We extended our analysis of Model 2 to four subsample data to examine the robustness of the main results presented in subsection 4.2.2. Table 28 displays the details. Panel (a) through panel (d) present four different segments of the sample data. Panel (a) is for 9,387 hotel-month data whose HasMR is positive, which means in the month the hotel ever responded to at least one review. Panel (b) is for 7,000 hotel-month data in which each hotel has continuous records for each month from January 2017 to December 2019. Panel (c) is for 8,533 hotel-month data in which the class of the hotels is 3-star or above, not bottom class. Panel (d) is for 4,091 hotel-month data in which the class of the hotels is below 3-star, that is bottom class. The reason we chose these four segments is that each of them represents one type of context. For instance, in the first subsample, HasMR ratio is greater than zero, it means this hotel this month shows the sense and effort in responding consumer reviews; the second subsample means this hotel is operated continuously not for just one time or a short time; the third and the fourth sub-sample represent instances of two different hotel class categories, not bottom and bottom, respectively.

For each panel, there are two columns, the first one contains variables of interest only, and the second includes control variables. As we can see, most of the significance and signs are the same as in Table 26 with a few exceptions, meaning our findings are solid. We will discuss

the exceptions in the next subsection.

Similarly, we extended and checked our findings of Model 3 to the same above four subsegments. The results are shown in Table 29. Also, the main results and findings of Model 3 are held by comparing Table 29 to Table 27. We will discuss a few exceptions in the next subsection.

Table 28 Robustness check of Model 2

Dependent Variable:	**MonthlyRating**							
	(a)		**(b)**		**(c)**		**(d)**	
	Model 2.5	**Model 2.6**	**Model 2.7**	**Model 2.8**	**Model 2.9**	**Model 2.10**	**Model 2.11**	**Model 2.12**
Estimator	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS
No. Observations	9387	9387	7000	7000	8533	8533	4091	4091
Cov. Est.	Robust	Robust	Robust	Robust	Robust	Robust	Robust	Robust
R-squared	0.0820	0.1185	0.0073	0.0695	0.0595	0.1092	0.0464	0.0828
R-Squared (Within)	0.0820	0.1185	0.0073	0.0695	0.0595	0.1092	0.0464	0.0828
R-Squared (Between)	0.0590	0.1919	-0.0109	0.0520	-0.0235	0.0794	-0.0398	0.1119
R-Squared (Overall)	0.0514	0.1225	-0.0030	0.0597	0.0230	0.0974	0.0157	0.0997
F-statistic	100.30	92.918	6.2673	39.022	65.146	77.712	23.814	27.112
P-value (F-stat)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Independent Variables:								
Fake	0.1101	0.0619	0.2108***	0.1545**	0.1508*	0.1138	0.1455	0.0939
	(0.0736)	(0.0728)	(0.0737)	(0.0696)	(0.0823)	(0.0802)	(0.0890)	(0.0876)
NFake	0.1026**	0.1871***	0.0007	0.0914*	0.1305**	0.1865***	0.1638***	0.2205***
	(0.0468)	(0.0481)	(0.0522)	(0.0515)	(0.0513)	(0.0518)	(0.0576)	(0.0588)
MRs Variables:								
HasMR	-0.1275***	-0.0938***	-0.0409	-0.0380	-0.0503	-0.0660**	0.3121***	0.2966***
	(0.0316)	(0.0304)	(0.0286)	(0.0273)	(0.0334)	(0.0324)	(0.0743)	(0.0718)
MR_WordCnt	-0.0064***	-0.0057***	-1.381e-05**	-1.771e-05***	-0.0041***	-0.0037***	-0.0075***	-0.0071***
	(0.0004)	(0.0004)	(6.579e-06)	(6.59e-06)	(0.0003)	(0.0003)	(0.0007)	(0.0007)
TextSimilarity	1.4034***	1.5548***	0.0223***	0.0260***	1.5012***	1.5308***	1.8339***	2.0140***
	(0.2748)	(0.2814)	(0.0075)	(0.0073)	(0.2530)	(0.2557)	(0.6479)	(0.6725)
Lag MRs Variables:								
HasMR_p1	0.0946***	0.1038***	0.1025***	0.1137***	0.0997***	0.1213***	0.1078	0.0989
	(0.0309)	(0.0302)	(0.0269)	(0.0256)	(0.0310)	(0.0303)	(0.0690)	(0.0671)
MR_WordCnt_p1	0.0017***	0.0015***	-5.165e-06	-8.335e-06	0.0009***	0.0007***	0.0016***	0.0015***
	(0.0003)	(0.0003)	(6.051e-06)	(5.942e-06)	(0.0002)	(0.0002)	(0.0006)	(0.0006)
TextSimilarity_p1	-0.0582	-0.0684	0.0168*	0.0193**	0.1809	0.1904	-0.7244	-0.8184*
	(0.2501)	(0.2476)	(0.0088)	(0.0084)	(0.2159)	(0.2147)	(0.5073)	(0.4930)
Control Variables:								
HasPhoto		-0.1411*		0.0128		0.0719		-0.1980**
		(0.073)		(0.096)		(0.0739)		(0.0807)
RevWordCnt		-0.0010***		-0.0015***		-0.0015***		-0.0011***
		(0.000)		(0.000)		(0.0003)		(0.0003)

Rev_I_Cnt		-0.0446***		-0.0591***		-0.0456***		-0.0374***
		(0.010)		(0.012)		(0.0102)		(0.0108)
TripTypeBusiness		-0.1156**		-0.1638***		-0.0848		-0.1418*
		(0.054)		(0.052)		(0.0517)		(0.0836)
LnMonthlyReviewsCount		0.0644***		0.0382***		0.0512***		0.0917***
		(0.014)		(0.014)		(0.0140)		(0.0264)
Const	3.9477***	3.9468***	4.0771***	4.0765***	4.0539***	4.0529***	3.3633***	3.3610***
	(0.0054)	(0.0053)	(0.0048)	(0.0046)	(0.0052)	(0.0051)	(0.0130)	(0.0128)
Effects	Entity	Entity	Entity	Entity	Entity	Entity	Entity	Entity

Note: Coefficients are shown in the table; standard errors are reported in parentheses.

*p<0.10; **p<0.05; ***p<0.01

Table 29 Robustness check of Model 3

Dependent Variable:	**MonthlyRatingGR**							
	(a)		**(b)**		**(c)**		**(d)**	
	Model 3.5	**Model 3.6**	**Model 3.7**	**Model 3.8**	**Model 3.9**	**Model 3.10**	**Model 3.11**	**Model 3.12**
Estimator	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS	PanelOLS
No. Observations	9387	9387	7000	7000	8533	8533	4091	4091
Cov. Est.	Robust	Robust	Robust	Robust	Robust	Robust	Robust	Robust
R-squared	0.4697	0.4786	0.4273	0.4484	0.4620	0.4804	0.4523	0.4619
R-Squared (Within)	0.4697	0.4786	0.4273	0.4484	0.4620	0.4804	0.4523	0.4619
R-Squared (Between)	-1.0789	-1.0148	-34.458	-33.315	-4.3913	-4.0337	-1.7224	-1.5640
R-Squared (Overall)	0.2048	0.2235	-0.0343	0.0017	0.1740	0.2126	0.2162	0.2428
F-statistic	884.27	589.02	562.90	394.10	786.35	543.92	358.76	239.40
P-value (F-stat)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Independent Variables:								
Fake	0.0068	-0.0050	0.0542*	0.0409	0.0611	0.0486	0.0119	-0.0059
	(0.0376)	(0.0376)	(0.0306)	(0.0299)	(0.0380)	(0.0377)	(0.0500)	(0.0498)
NFake	0.0128	0.0409	0.0094	0.0325*	0.0567**	0.0753***	0.0695**	0.0911***
	(0.0243)	(0.0253)	(0.0191)	(0.0191)	(0.0242)	(0.0241)	(0.0337)	(0.0347)
MRs Variables:								
HasMR	-0.0246	-0.0168	0.0019	0.0023	-0.0157	-0.0228	0.1489***	0.1401***
	(0.0161)	(0.0153)	(0.0105)	(0.0101)	(0.0175)	(0.0174)	(0.0388)	(0.0384)
MR_WordCnt	-0.0022***	-0.0020***	-3.021e-06*	-3.189e-06*	-0.0012***	-0.0010***	-0.0031***	-0.0029***
	(0.0002)	(0.0002)	(1.77e-06)	(1.756e-06)	(0.0001)	(0.0001)	(0.0003)	(0.0003)

TextSimilarity	0.6728***	0.7118***	0.0053**	0.0066***	0.6107***	0.6201***	1.1201***	1.2014***
	(0.1432)	(0.1447)	(0.0021)	(0.0020)	(0.1510)	(0.1507)	(0.2972)	(0.3076)
Lag MRs Variables:								
HasMR_p1	0.0164	0.0188	0.0171	0.0199*	0.0376**	0.0452***	0.0561	0.0524
	(0.0169)	(0.0167)	(0.0116)	(0.0112)	(0.0170)	(0.0171)	(0.0390)	(0.0389)
MR_WordCnt_p1	0.0006***	0.0005***	-2.137e-06	-2.417e-06	0.0002	0.0001	0.0004	0.0004
	(0.0002)	(0.0002)	(1.898e-06)	(1.846e-06)	(0.0001)	(0.0001)	(0.0004)	(0.0004)
TextSimilarity_p1	-0.1822	-0.1764	0.0033	0.0040	-0.0068	-0.0018	-0.5399*	-0.5668**
	(0.1159)	(0.1169)	(0.0028)	(0.0025)	(0.1002)	(0.1005)	(0.2830)	(0.2830)
Lag Rating Variable:								
MonthlyRating_p1	-0.4126***	-0.4142***	-0.3041***	-0.3064***	-0.3958***	-0.3954***	-0.4898***	-0.4909***
	(0.0145)	(0.0143)	(0.0143)	(0.0139)	(0.0173)	(0.0170)	(0.0152)	(0.0150)
Control Variables:								
HasPhoto		-0.0839***		-0.0226		0.0286		-0.0817*
		(0.032)		(0.034)		(0.0393)		(0.0448)
RevWordCnt		-0.0002		-0.0004***		-0.0006***		-0.0005***
		(0.000)		(0.000)		(0.0001)		(0.0002)
Rev_I_Cnt		-0.0150**		-0.0166***		-0.0144***		-0.0132*
		(0.006)		(0.004)		(0.0044)		(0.0069)
TripTypeBusiness		-0.0587**		-0.0507***		-0.0428*		-0.0550
		(0.025)		(0.018)		(0.0226)		(0.0446)
LnMonthlyReviewsCount		0.0113		-0.0030		0.0068		0.0136
		(0.007)		(0.006)		(0.0064)		(0.0140)
Const	0.0330***	0.0328***	0.0123***	0.0120***	0.0263***	0.0260***	0.1148***	0.1140***
	(0.0026)	(0.0026)	(0.0017)	(0.0016)	(0.0023)	(0.0023)	(0.0072)	(0.0072)
Effects	Entity	Entity	Entity	Entity	Entity	Entity	Entity	Entity

Note: Coefficients are shown in the table; standard errors are reported in parentheses.

*p<0.10; **p<0.05; ***p<0.01

4.2.4 Hypotheses testing

As we claim in section 4.1, all five hypotheses tested by Model 1 hold, four out of five hypotheses tested by Model 2 and Model 3, respectively, also get supported. Only the two hypotheses about the negative association between fake review ratio and hotel rating or its growth rate get no support, and we get some opposite evidence instead.

In this section, we focus on the few exceptions about Model 2 and Model 3. Table 30 and Table 31 show a combination of the main results and robustness check for each Model with only the signs of the coefficients but omitting specific values, and highlighting a few exceptions, plus the first two columns showing whether the hypothesis holds. The cells with green shading indicate the unexpected negative coefficient with significance, the light brown cells denote unexpected positive coefficient, and the grey indicates insignificance which is inconsistent with the main results shown in Model 2.4 or Model 3.4.

When examining the effect of three MR variables, we recall we should use a collective approach and check the magnitude if necessary.

First, let's look at the HasMR variable. Although we have several unexpected negative or positive or insignificance in the robustness check of both models, we find that the main findings, which is positive association between HasMR and rating or rating growth rate, still holds after checking the magnitude of each effect. For instance, Model 2.6 in Table 30, we have an unexpected negative contemporaneous effect, but since its magnitude is -.0938 is smaller than the positive one lag effect, which is .1038, the collective effect will still be positive, so that the H2c holds. Similar analysis can be done for Model 2.10, 2.12 and Model 3.12 in Table 31. But Model 3.6 shows HasMR has no significant impact even from a collective perspective. This is a real exception. We will discuss this situation in section 5.

The remaining unexpected and highlighted entries about MRWordCnt and

TextSimilarity in Table 30 and Table 31 can be easily explained if looking in a collective way. Therefore, H2/3d and H2/3e hold still.

Table 30 Hypotheses testing results: Model 2

Model 2:								
Dep. Variable	**Ratings**		**Main results**		**Robustness check**			
	HYPOS	Testing	all	all	HasMR > 0	Complete	Mid and above hotels	Bottom class hotels
			Model 2.2	**Model 2.4**	**Model 2.6**	**Model 2.8**	**Model 2.10**	**Model 2.12**
No. Observations			+	+	n.s.	+	n.s.	n.s.
Rsquared			+	+	+	+	+	+
Independent Variables:								
Fake	H2a	×	+	n.s.	-	n.s.	-	+
NFake	H2b	√	-	-	-	-	-	-
MRs Variables:								
HasMR	H2c	√	+	n.s.	-	n.s.	-	+
MR_WordCnt	H2d	√	-	-	-	-	-	-
TextSimilarity	H2e	√	+	+	+	+	+	+
Lag MRs Variables:								
HasMR_p1				+	+	+	+	n.s.
MR_WordCnt_p1				+	+	n.s.	+	+
TextSimilarity_p1				n.s.	n.s.	+	n.s.	-

Table 31 Hypotheses testing results: Model 3

Model 3:								
Dep. Variable	**Ratings GR**		**Main results**		**Robustness check**			
	HYPOS	Testing	all	all	HasMR > 0	Complete	Mid- and above hotels	Bottom class hotels
			Model 3.2	**Model 3.4**	**Model 3.6**	**Model 3.8**	**Model 3.10**	**Model 3.12**
No. Observations			12624	12624	9387	7000	8533	4091
Rsquared			0.4584	0.4591	0.4786	0.4484	0.4804	0.4619
Independent Variables:								
Fake	H3a	×	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
NFake	H3b	√	+	+	n.s.	+	+	+
MRs Variables:								
HasMR	H3c	√	+	n.s.	n.s.	n.s.	n.s.	+
MR_WordCnt	H3d	√	-	-	-	-	-	-
TextSimilarity	H3e	√	+	+	+	+	+	+
Lag MRs Variables:								-
HasMR_p1				+	n.s.	+	+	n.s.
MR_WordCnt_p1				n.s.	+	n.s.	n.s.	n.s.
TextSimilarity_p1				n.s.	n.s.	n.s.	n.s.	-

5 Discussion and implication

5.1 Discussion

Based on a dataset of 366,417 reviews from Tripadvisor.com and a derived and aggregated hotel-month panel dataset (involving 392 hotels and 36 months, in total 13,081 instances), this study empirically examined two major interplays of fake review detection, managerial response, hotel rating and its growth. To test our hypotheses, we built three econometric models. The first one is a logistic regression model, aiming to disclose the association between multiple attributes of review and the chance for a review to receive a response. Both the second and the third are fixed effect OLS regression models, which catch the effects of review and response on eWOM. In general, this study reached the following conclusions:

First, by employing a self-made fake online review classifier, we labeled each review in the sample as fake, non-fake, or undecided. We found being attached to a fake or a non-fake tag has an opposite impact on the likelihood of a review receiving a response. On average, in contrast to an undecided review, being fake leads to 2.81% less likely, while non-fake means 2.55% more likely. Furthermore, being congruent, a review increases the likelihood of getting an MR by 11.8%. And the probability increases by 9.8% for a review with one point more deviation. But, for a review about a top-class hotel, the chance decreases by 1.52%.

Second, with a deeper understanding on managerial response and fake review detection, we further examine the correlation between them and hotel rating and its growth. Based on the entire panel dataset and several meaningful subsamples, we found almost consensually that a high percentage of non-fake reviews, relatively high managerial response rate, and a concise but matching response content are beneficial to both the levels and growth of hotel rating.

But we did not find evidence showing a high percentage of fake reviews is harmful to hotel rating itself or its growth, on the contrary we noticed that the variable positively correlates with hotel rating. One potential explanation would be a large portion of fake review is extremely positive and thus hotel rating is positively impacted. Seen from Table 18, on average 25% of fake reviews are 5-star, while 23% of fake review are 1-star. But we still need further evidence to be convinced of the reasoning behind.

In addition, from the robustness check of Model 3 on the first sub-sample, in which each hotel-month instance ever responded to review at least once, we found that the variable of MR ratio has no significant correlation with hotel rating growth, though the variable is significantly positive correlated with the level of hotel rating. This real exception offers a great perspective to us to understand deeper about the role of managerial response. For hotels which already employed the response strategy either from consideration of service failure recovery, showing care and gratitude to customers, or outreaching proactively to reinforce good brand image 32, it maybe not worthwhile for them to maintain a very high level of response at all costs. The exception offers some evidence on the guess. In the next stage, after adopting the strategy, hotels maybe should pay closer attention to the questions about to which review, to which reviewer, how, and who responds (L. Wang et al., 2020; Zhu et al., 2021). Hotels need to reshape the strategy and evaluate when to get the most potential benefit from the MR strategy instead of spending too many resources on just maintaining a good responsive image.

5.2 Theoretical implications

This study contributes to the literature of managerial response in a few ways. First, it is one of the first to examine the association between fake review detection and managerial response. In terms of the source of variables, most prior studies had to use the available second-handed

information offered by websites or other outside data service provider, or design manipulated survey in a laboratory, but this study shows that by employing self-made identification algorithm, to some extent, researchers can break through the limit of current resource of data.

Second, it enriches the body of growing knowledge on employing natural language processing (NLP) technology in marketing area (Berger et al., 2020; Chang et al., 2020; Ma & Sun, 2020; Urban et al., 2020; X. Zhang et al., 2020). By making use of unstructured text analysis, calculating the cosine similarity between every pair of managerial response and consumer review, and examining its association with review rating and rating growth, this study shows that the method and metric derived from NLP method could help broaden marketing research by offering new insights, perspectives, and methods to research of interest.

Lastly, methodologically speaking this study introduces a finite distributed lag model (with the shortest possible one-term lag) to managerial response research area. The model discloses a perspective for researchers to see the dynamic impact of independent variables on dependent variables, from both contemptuous and lagged angles. Previous studies usually only caught the lag effect by using one month or one quarter lagged value of independent variables as inputs (K. L. Xie et al., 2016; X. Zhang et al., 2020).

5.3 Practical implications

This study also has multiple implications for both practicing managers and consumers. First, by examining the associations among online review, managerial response, and hotel rating, this study suggests that hotels should attract more truthful reviews from valid customers. As we can see, the percentage of non-fake reviews has a strong and positive association with hotel rating and its growth. As for fake reviews, though not much evidence showing committing review fraud is harmful to hotel rating itself, this study shows fake review ratio has no significant

effect on hotel rating, especially to hotels that have regularly responded to reviews.

Second, rather than responding to all reviews using an automatic response tablet, managers should carefully decide which review(er) to respond to and make responses be concise and matching while managers should be actively responding to reviews. Unlike most prior studies from which shows the worth of adopting MR strategy and positive effects of it on ratings and sales, this study shows the picture from another perspective. By examining the association between main aspects of managerial response with hotel rating and its growth, this study discloses achieving a high response rate is not always the right thing to do, such as for hotels who already began to respond to reviews regularly.

Third, increase the response rate to reviews from people who have many followers in contrast to those who have none or just a few followers, and make the response in an appropriate way, concise and matching. Considering that a larger size of followers usually leads to a higher exposure rate, it could be a good chance for hotels to reinforce their good fame or attenuate their failure to a larger size of audience. We could also make a similar increase in response to reviews that post photos along with the texts.

Finally, this study also shed light on consumers especially who want to their reviews get more attentions from hotels. A truthful, detailed, congruent, and deviated review is more likely to receive a response from hotel. But if the review is for a top-class hotel, the chance decreases.

5.4 Limitations and future research

We launched this study as a part of a bigger plan, which includes the sales performance information as one variable of interest and in which interconnection among sales, rating, review, and response is the research question. Though prior studies have done it in the hospitality, restaurants, and books industry before (Chevalier & Mayzlin, 2006; Huang et al.,

2021; K. L. Xie et al., 2016), they either used sales rank as a proxy, no mention, or got access to the sales data from a commercial company. But what we wanted to do is to find a way to get free, rich, and reliable data on both sales and response, as of today it is still under study. We will continue our research to dig deeper in this direction.

Fake and NFake identification is still relatively new and, to a great extent, lacks in way of verification. In this study, we employed a simple deep learning classifier, which was trained by data from Yelp.com showing better performance among multiple choices with 85% accuracy. Future scholars are welcome to join the research stream of suspicious review detection and transferred learning and testify the generalizability and validity of the classifier used in this study.

We collected data well after the reviews and responses got posted, and thus some interesting but dynamic attributes were unavailable to us, such as the exact date of each review, the change of hotel overall review rating and ranking, the visibility of response and review at real-time, and lots more. Though many prior studies suggest the importance of quick response (InsideSales.com, 2007)(Li et al., 2017; Sheng, 2019), we did not find significant relationship between the response speed and the hotel rating by using not exactly correct managerial response interval data. We are still interested in this topic since speedy response means much more resource must be spent and if speedy response is not very important in most cases, it will save a lot for hotels by employing a not-rush response strategy. Min et al. (2015) already found speed to negative reviews did not influence the ratings of the response though not on the hotel rating.

One promising direction would be to use multiple text similarity measurement methods rather than only use cosine similarity between vectors basing on TF-IDF scores to represent words and find out which way has a higher level of indication property.

Another direction is to examine the short-run temporary and long-run cumulative effects of managerial response on e-WOM and its change. One way could be to continue using the same finite distributed lag model, which is suitable to estimating dynamic relationships between variables (J. Parker), as what we did in this study, but try larger lag length rather than just one period lag and employ some restrictions on the lag distribution to impose smooth lag weights. Commonly used restricted distributed lags such as linearly declined and polynomial distributed lag, which was first explored by Almon (1965), may be applied.

References

Almon, S. (1965). The Distributed Lag Between Capital Appropriations and Expenditures. *Econometrica*, *33*(1), 178–196. https://doi.org/10.2307/1911894

ANDERSON, E. T., & SIMESTER, D. I. (2014). Reviews Without a Purchase: Low Ratings, Loyal Customers, and Deception. *Journal of Marketing Research*, *51*(3), 249–269.

Aral, S. (2014). *The Problem With Online Ratings*. MIT Sloan Management Review. Retrieved May 10, 2021, from https://sloanreview.mit.edu/article/the-problem-with-online-ratings-2/

Babić Rosario, A., Sotgiu, F., De Valck, K., & Bijmolt, T. H. A. (2016). The Effect of Electronic Word of Mouth on Sales: A Meta-Analytic Review of Platform, Product, and Metric Factors. *Journal of Marketing Research*, *53*(3), 297–318. https://doi.org/10.1509/jmr.14.0380

Berger, J., Humphreys, A., Ludwig, S., Moe, W. W., Netzer, O., & Schweidel, D. A. (2020). Uniting the Tribes: Using Text for Marketing Insight. *Journal of Marketing*, *84*(1), 1–25. https://doi.org/10.1177/0022242919873106

Chang, Y.-C., Ku, C.-H., & Chen, C.-H. (2020). Using deep learning and visual analytics to explore hotel reviews and responses. *Tourism Management*, *80*, 104129. https://doi.org/10.1016/j.tourman.2020.104129

Chen, W., Gu, B., Ye, Q., & Zhu, K. X. (2019). Measuring and Managing the Externality of Managerial Responses to Online Customer Reviews. *Information Systems Research*, *30*(1), 81–96. https://doi.org/10.1287/isre.2018.0781

Chevalier, J. A., Dover, Y., & Mayzlin, D. (2018). Channels of Impact: User Reviews When Quality Is Dynamic and Managers Respond. *Marketing Science*, *37*(5), 688–709. https://doi.org/10.1287/mksc.2018.1090

Chevalier, J. A., & Mayzlin, D. (2006). The Effect of Word of Mouth on Sales: Online Book Reviews. *Journal of Marketing Research*, *43*(3), 345–354.

Connelly, B., Certo, T., Ireland, R., & Reutzel, C. (2011). Signaling Theory: A Review and Assessment. *Journal of Management - J MANAGE*, *37*, 39–67. https://doi.org/10.1177/0149206310388419

Esmark Jones, C. L., Stevens, J. L., Breazeale, M., & Spaid, B. I. (2018). Tell it like it is: The effects of differing responses to negative online reviews: ESMARK JONES ET AL. *Psychology & Marketing*, *35*(12), 891–901. https://doi.org/10.1002/mar.21142

Folkes, V. S. (1988). Recent Attribution Research in Consumer Behavior: A Review and New Directions. *Journal of Consumer Research*, *14*(4), 548–565. https://doi.org/10.1086/209135

Groth, M., Hennig-Thurau, T., & Walsh, G. (2009). Customer Reactions to Emotional Labor: The Roles of Employee Acting Strategies and Customer Detection Accuracy. *Academy of Management Journal*, *52*(5), 958–974. https://doi.org/10.5465/AMJ.2009.44634116

Gu, B., & Ye, Q. (2014). First Step in Social Media: Measuring the Influence of Online Management Responses on Customer Satisfaction. *Production & Operations Management*, *23*(4), 570–582. https://doi.org/10.1111/poms.12043

Hausman, J. A. (1978). Specification Tests in Econometrics. *Econometrica*, *46*(6), 1251–1271. https://doi.org/10.2307/1913827

He, S., Hollenbeck, B., & Proserpio, D. (2020). *The Market for Fake Reviews* (SSRN Scholarly Paper ID 3664992). Social Science Research Network. https://doi.org/10.2139/ssrn.3664992

Hogreve, J., Bilstein, N., & Mandl, L. (2017). Unveiling the recovery time zone of tolerance: When time matters in service recovery. *Journal of the Academy of Marketing Science*, *45*(6), 866–883. http://dx.doi.org.ezproxy.lib.uwm.edu/10.1007/s11747-017-0544-7

Huang, Y., Jin, Y., & Huang, J. (2021). Impact of Managerial Responses on Product Sales: Examining the Moderating Role of Competitive Intensity and Market Position. *Journal of the Association for Information Systems*, *22*(2), 1. http://dx.doi.org.ezproxy.lib.uwm.edu/10.17705/1jais.00671

Israeli, A. A. (2002). Star rating and corporate affiliation: Their influence on room price and performance of hotels in Israel. *International Journal of Hospitality Management*, *21*(4), 405–424. https://doi.org/10.1016/S0278-4319(02)00037-3

Kelley, H. H., & Michela, J. L. (1980). Attribution Theory and Research. *Annual Review of Psychology*, *31*(1), 457–501. https://doi.org/10.1146/annurev.ps.31.020180.002325

Kwok, L., & Xie, K. L. (2016). Factors contributing to the helpfulness of online hotel reviews. *International Journal of Contemporary Hospitality Management*, *28*(10), 2156–2177. http://dx.doi.org.ezproxy.lib.uwm.edu/10.1108/IJCHM-03-2015-0107

Lappas, T., Sabnis, G., & Valkanas, G. (2016). The Impact of Fake Reviews on Online Visibility: A Vulnerability Assessment of the Hotel Industry. *Information Systems Research*, *27*(4), 940–961. https://doi.org/10.1287/isre.2016.0674

Li, C., Cui, G., & Peng, L. (2017). The signaling effect of management response in engaging customers: A study of the hotel industry. *Tourism Management*, *62*, 42–53. https://doi.org/10.1016/j.tourman.2017.03.009

Liang, S., & Li, H. (2019). Respond More to Good Targets: An Empirical Study of Managerial Response Strategy in Online Travel Websites. *E-Review of Tourism Research*, *16*(2/3), Article 2/3. https://ertr-ojs-tamu.tdl.org/ertr/index.php/ertr/article/view/334

Luca, M. (2016). *Reviews, Reputation, and Revenue: The Case of Yelp.Com* (SSRN Scholarly Paper ID 1928601). Social Science Research Network. https://doi.org/10.2139/ssrn.1928601

Luca, M., & Zervas, G. (2016). Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud. *Management Science*, *62*(12), 3412–3427. https://doi.org/10.1287/mnsc.2015.2304

Ma, L., & Sun, B. (2020). Machine learning and AI in marketing – Connecting computing power to human insights. *International Journal of Research in Marketing*, *37*(3), 481–504. https://doi.org/10.1016/j.ijresmar.2020.04.005

Mayzlin, D., Dover, Y., & Chevalier, J. (2014). Promotional Reviews: An Empirical Investigation of Online Review Manipulation. *American Economic Review*, *104*(8), 2421–2455. https://doi.org/10.1257/aer.104.8.2421

McCollough, M. A., Berry, L. L., & Yadav, M. S. (2000). An empirical investigation of customer satisfaction after service failure and recovery. *Journal of Service Research : JSR*, *3*(2), 121–137.

Min, H., Lim, Y., & Magnini, V. P. (2015). Factors Affecting Customer Satisfaction in Responses to Negative Online Hotel Reviews: The Impact of Empathy, Paraphrasing, and

Speed. *Cornell Hospitality Quarterly*, *56*(2), 223–231. https://doi.org/10.1177/1938965514560014

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013). *What Yelp Fake Review Filter Might Be Doing?* http://www2.cs.uh.edu/~arjun/papers/ICWSM-Spam_final_camera-submit.pdf

Osgood, C. E., & Tannenbaum, P. H. (1955). The principle of congruity in the prediction of attitude change. *Psychological Review*, *62*(1), 42–55. https://doi.org/10.1037/h0048153

Proserpio, D., & Zervas, G. (2017). Online Reputation Management: Estimating the Impact of Management Responses on Consumer Reviews. *Marketing Science*, *36*(5), 645–665. https://doi.org/10.1287/mksc.2017.1043

Sheng, J. (2019). Being Active in Online Communications: Firm Responsiveness and Customer Engagement Behaviour. *Journal of Interactive Marketing*, *46*, 40–51. https://doi.org/10.1016/j.intmar.2018.11.004

Spreng, R. A., Harrell, G. D., & Mackoy, R. D. (1995). Service recovery: Impact on satisfaction and intentions. *The Journal of Services Marketing*, *9*(1), 15. http://dx.doi.org.ezproxy.lib.uwm.edu/10.1108/08876049510079853

Surachartkumtonkun, J. (Nui), Grace, D., & Ross, M. (2020). Unfair customer reviews: Third-party perceptions and managerial responses. *Journal of Business Research*, S0148296320306494. https://doi.org/10.1016/j.jbusres.2020.09.071

Updating my hotel star rating. (n.d.). TripAdvisor Help Center. Retrieved May 24, 2021, from https://www.tripadvisorsupport.com/hc/en-us/articles/200614057-Updating-my-hotel-star-rating

Urban, G., Timoshenko, A., Dhillon, P., & Hauser, J. R. (2020). Is Deep Learning a Game Changer for Marketing Analytics? *MIT Sloan Management Review; Cambridge*, *61*(2), 70–76.

Wang, L., Ren, X., Wan, H., & Yan, J. (2020). Managerial responses to online reviews under budget constraints: Whom to target and how. *Information & Management*, *57*(8), 103382. https://doi.org/10.1016/j.im.2020.103382

Wang, Y., & Chaudhry, A. (2015). *The Effect of Observing the Service Interaction of Peers: Empirical Evidence of the Pros and Cons to Responding to Online Customer Reviews* (SSRN Scholarly Paper ID 2630237). Social Science Research Network. https://doi.org/10.2139/ssrn.2630237

Wang, Y., & Chaudhry, A. (2018). When and how Managers' Responses to Online Reviews Affect Subsequent Reviews. *Journal of Marketing Research*, *55*(2), 163–177. https://doi.org/10.1509/jmr.15.0511

Wei, W., Miao, L., & Huang, Z. (Joy). (2013). Customer engagement behaviors and hotel responses. *International Journal of Hospitality Management*, *33*, 316–330. https://doi.org/10.1016/j.ijhm.2012.10.002

Xie, K., Kwok, L., & Wang, W. (2017). Monetizing Managerial Responses on TripAdvisor: Performance Implications Across Hotel Classes. *Cornell Hospitality Quarterly*, *58*(3), 240–252. https://doi.org/10.1177/1938965516686109

Xie, K. L., & So, K. K. F. (2018). The Effects of Reviewer Expertise on Future Reputation, Popularity, and Financial Performance of Hotels: Insights from Data-Analytics. *Journal of Hospitality & Tourism Research*, *42*(8), 1187–1209. https://doi.org/10.1177/1096348017744016

Xie, K. L., So, K. K. F., & Wang, W. (2017). Joint effects of management responses and online reviews on hotel financial performance: A data-analytics approach. *International Journal of Hospitality Management*, *62*, 101–110. https://doi.org/10.1016/j.ijhm.2016.12.004

Xie, K. L., Zhang, Z., Zhang, Z., Singh, A., & Lee, S. K. (2016). Effects of managerial response on consumer eWOM and hotel performance. *International Journal of Contemporary Hospitality Management*, *28*(9), 2013–2034. http://dx.doi.org.ezproxy.lib.uwm.edu/10.1108/IJCHM-06-2015-0290

Xu, Y., Li, H., Law, R., & Zhang, Z. (2020). Can receiving managerial responses induce more user reviewing effort? A mixed method investigation in hotel industry. *Tourism Management*, *77*, 103982. https://doi.org/10.1016/j.tourman.2019.103982

Xu, Y., Zhang, Z., Law, R., & Zhang, Z. (2020). Effects of online reviews and managerial responses from a review manipulation perspective. *Current Issues in Tourism*, *23*(17), 2207–2222. https://doi.org/10.1080/13683500.2019.1626814

Zhang, X., Qiao, S., Yang, Y., & Zhang, Z. (2020). Exploring the impact of personalized management responses on tourists' satisfaction: A topic matching perspective. *Tourism Management*, *76*, 103953. https://doi.org/10.1016/j.tourman.2019.103953

Zhang, Z., Li, H., Meng, F., & Li, Y. (2019). The effect of management response similarity on online hotel booking: Field evidence from Expedia. *International Journal of Contemporary Hospitality Management*, *31*(7), 2739–2758. https://doi.org/10.1108/IJCHM-09-2018-0740

Zhao, Y., Wen, L., Feng, X., Li, R., & Lin, X. (2020). How managerial responses to online reviews affect customer satisfaction: An empirical study based on additional reviews.

Journal of Retailing and Consumer Services, *57*, 102205. https://doi.org/10.1016/j.jretconser.2020.102205

Zheng, T., Wu, F., Law, R., Qiu, Q., & Wu, R. (2021). Identifying unreliable online hospitality reviews with biased user-given ratings: A deep learning forecasting approach. *International Journal of Hospitality Management*, *92*, 102658. https://doi.org/10.1016/j.ijhm.2020.102658

Zhu, J. J., Chang, Y.-C., Ku, C.-H., Li, S. Y., & Chen, C.-J. (2021). Online critical review classification in response strategy and service provider rating: Algorithms from heuristic processing, sentiment analysis to deep learning. *Journal of Business Research*, *129*, 860–877. https://doi.org/10.1016/j.jbusres.2020.11.007

Tripadvisor (2021), "2020 Annual Report", available at: https://ir.tripadvisor.com/static-files/fef1a79b-0b14-40b3-ae35-da7ee030aca4 (accessed 27 May 2021).

InsideSales (2007), "Lead Management Survey", available at: http://www.leadresponsemanagement.org/images/lrm-survey.pdf (accessed 29 May 2021).

J. Parker, "Chapter 3: Distributed-Lag Models", available at: https://www.reed.edu/economics/parker/312/tschapters/S13_Ch_3.pdf (accessed 29 May 2021)

Appendix

Table 33 Sample distribution 2017 - 2019

2017-2019														
Hotel	**Review**								**MR**					
Class	**Rating**	**Volume**	**Volume Ratio**	**Useful**	**Photo**	**Fake**	**NFake**	**Word Cnt**	**MR Volume**	**MR Ratio**	**Interval**	**General Mgr**	**Mgr**	**Word Cnt**
Bottom	1	2310	10%	23%	19%	24%	30%	140	1,409	61%	23	17%	15%	91
	2	1850	8%	14%	13%	20%	40%	127	1,203	65%	23	18%	17%	92
	3	3543	15%	8%	7%	18%	49%	116	2,232	63%	24	15%	17%	82
	4	6522	28%	8%	8%	18%	51%	104	4,044	62%	24	15%	14%	68
	5	9234	39%	8%	8%	25%	36%	87	6,556	71%	24	16%	13%	59
Bottom	Subtotal	23,459	100%	10%	9%	22%	42%	104	15,443	66%	24	16%	14%	70
Middle	1	10024	9%	25%	25%	23%	27%	148	6,616	66%	20	38%	6%	83
	2	8774	8%	15%	15%	20%	36%	147	6,142	70%	20	33%	7%	81
	3	16491	15%	10%	9%	18%	46%	127	10,389	63%	20	33%	9%	75
	4	29791	27%	10%	9%	18%	49%	111	12,810	43%	22	20%	10%	62
	5	45594	41%	10%	10%	25%	36%	90	22,341	49%	24	14%	12%	56
Middle	Subtotal	110,674	100%	12%	11%	21%	40%	111	58,298	54%	22	23%	10%	66
Top	1	13101	6%	37%	37%	23%	29%	167	9,171	70%	19	29%	25%	100
	2	12782	6%	24%	24%	19%	37%	167	9,714	76%	20	26%	23%	98
	3	23702	11%	16%	16%	18%	46%	147	15,643	66%	20	27%	23%	90
	4	47643	22%	11%	11%	17%	52%	119	21,439	45%	21	20%	27%	76
	5	115256	54%	11%	11%	25%	37%	90	54,170	47%	20	14%	31%	68
Top	Subtotal	212,484	100%	14%	14%	22%	41%	112	110,138	54%	20	19%	28%	78
Total		346, 617	100%	13%	13%	22%	41%	111	384,636	53%	21	20%	21%	74

Table 34 Sample descriptive statistics in Jan 2017 and Dec 2019

Jan 2017

Hotel Class	Review Rating	Volume	Volume Ratio	Useful	Photo	Fake	NFake	Word Cnt	MR Volume	MR Ratio	Interval	General Mgr	Mgr	Word Cnt
Bottom	1	91	12%	5%	7%	29%	29%	144	46	51%	20	11%	9%	80
	2	62	8%	0%	3%	23%	35%	117	34	55%	17	9%	6%	70
	3	136	17%	1%	6%	12%	53%	117	68	50%	22	3%	12%	66
	4	241	30%	0%	6%	18%	49%	105	149	62%	25	10%	5%	60
	5	261	33%	0%	3%	25%	38%	90	176	67%	23	15%	2%	56
Bottom	Subtotal	791	100%	1%	5%	21%	41%	115	473	61%	21	10%	7%	66
Middle	1	138	7%	9%	7%	17%	38%	178	102	74%	23	37%	10%	91
	2	134	7%	1%	4%	10%	49%	164	97	72%	24	34%	11%	91
	3	307	15%	2%	8%	10%	62%	142	219	71%	23	35%	10%	85
	4	617	31%	1%	6%	12%	63%	125	355	58%	25	23%	10%	68
	5	809	40%	0%	9%	19%	46%	108	511	63%	32	16%	14%	61
Middle	Subtotal	2,005	100%	3%	7%	14%	52%	143	1,284	65%	25	29%	11%	79
Top	1	135	3%	8%	10%	15%	47%	220	112	83%	21	20%	20%	100
	2	187	5%	1%	6%	13%	50%	218	162	87%	18	23%	23%	102
	3	430	11%	2%	7%	13%	62%	184	325	76%	20	32%	26%	99
	4	1,026	27%	1%	10%	11%	61%	142	699	68%	21	23%	27%	80
	5	2,093	54%	1%	10%	17%	49%	108	1,429	68%	20	14%	35%	72
Top	Subtotal	3,871	100%	3%	9%	14%	54%	174	2,727	71%	20	22%	26%	91
Total		6,667	100%	2%	8%	15%	52%	158	4,484	67%	22	23%	20%	85

Dec 2019

Hotel Class	Review Rating	Volume	Volume Ratio	Useful	Photo	Fake	NFake	Word Cnt	MR Volume	MR Ratio	Interval	General Mgr	Mgr	Word Cnt
Bottom	1	49	12%	4%	16%	22%	31%	154	24	49%	17	21%	12%	88
	2	34	8%	12%	12%	9%	47%	126	23	68%	20	13%	26%	106
	3	47	11%	6%	2%	15%	49%	109	32	68%	17	16%	12%	77
	4	90	21%	7%	4%	17%	47%	108	44	49%	19	18%	16%	60
	5	199	47%	6%	4%	22%	32%	83	144	72%	19	18%	16%	59
Bottom	Subtotal	419	99%	7%	8%	17%	41%	116	267	64%	18.4	17%	16%	78
Middle	1	299	11%	31%	9%	26%	22%	141	120	40%	15	3%	4%	75
	2	229	9%	21%	6%	20%	36%	146	121	53%	17	10%	8%	68
	3	353	13%	10%	7%	22%	36%	116	183	52%	17	23%	8%	66
	4	595	23%	8%	6%	24%	38%	95	160	27%	18	10%	8%	53
	5	1141	44%	7%	7%	32%	26%	77	445	39%	18	13%	8%	51
Middle	Subtotal	2,617	100%	15%	7%	25%	32%	115	1,029	39%	17	12%	7%	62.6

Top	1	494	9%	44%	7%	24%	23%	152	176	36%	17	3%	32%	100
	2	381	7%	35%	6%	18%	31%	138	180	47%	18	4%	24%	94
	3	647	12%	23%	6%	23%	33%	123	287	44%	17	26%	20%	75
	4	987	18%	15%	8%	24%	40%	100	252	26%	19	14%	30%	71
	5	3,076	55%	13%	8%	32%	28%	76	1,016	33%	20	13%	31%	76
Top	Subtotal	5,585	100%	26%	7%	24%	31%	118	1,911	34%	18.2	12%	27%	83
Total		8,621	100%	22%	7%	24%	32%	117	3,207	37%	17.8	12%	20%	76

Curriculum Vitae

Long Chen

EDUCATION

Ph.D. in Management Science, Lubar School of Business, University of Wisconsin-Milwaukee 2017-2021(expected)
Concentration: Marketing
Minor: Econometrics
Ph.D. in Accounting, Xiamen University 1999-2002
Concentrations: Management Accounting
M.S. in Accounting, Xiamen University 1996-1999
B.S. in Chemical Engineering, Hainan University 1992-1996

RESEARCH INTERESTS

Online reputation management, Online reviews, Fake review detection, Managerial responses, Application of machine learning and deep learning methods, and Interdisciplinary research.

PEER-REVIEWED PUBLICATIONS

Huang. Z., and **Chen. L.,** (2000), "An Empirical Study on Earnings Growth Pattern of Public Companies in China", *Journal of Economics Research,* (in Chinese)
Chen. L., and Liu, Z., (2002), "The Real Option Value in Project Evaluation", SiChuan Accounting Journal, (in Chinese)
Chen. L., and Liu, Z., (1998), "Comments on the Substance over Form Principle", The Journal of Finance and Accounting, (in Chinese)

WORKING PAPERS

Chen. L., Ghose, S., Bhatnagar, A. and Liu. Z., Fake Online Review Detection Classifiers: A Comparative Study. In preparation for submission.
Chen. L., Ghose, S., Bhatnagar, A. and Liu. Z., Online Reviews, Managerial Responses, and Hotel Ratings: Evidence from Tripadvisor. In preparation for submission.

TEACHING

University Wisconsin-Milwaukee, Lubar School of Business
Instructor: Undergraduate Students
Internet Marketing, 2 sections, Online, Fall 2020 - Spring 2021
Internet Marketing, 3 sections, Hybrid (online and in-class), Spring 2020
Internet Marketing, 2 sections, In-Class, Fall 2019

Teaching Assistant:
Principle of Marketing (Dr. Grace Ambrose), Fall 2018 – Spring 2019

Xiamen University
Instructor: College of Continuous Education, Intermediate Accountant Course Trainee
Financial Management, 1 section, Fall 2000 – Spring 2001
Teaching Assistant: The Department of Accounting, Undergraduate Students
Auditing (Dr. Guirong Guo), 1 section, Fall 1998

AWARDS

Outstanding Doctoral Student Teaching Award, Lubar School of Business, UWM, 2021
UWM Fellowship, University of Wisconsin-Milwaukee, 2017-2021
The Third Prize of University Humanities and Social Sciences Outstanding Achievements, the Ministry of Education (China), 2002
Full Scholarship, Xiamen University, 1996-2002
First-Class Scholarship, Hainan University, 1992-1996

CERTIFICATES

CPA (Certified Public Accountant, China), since 1999
Deep Learning Specialization Certificate, Coursera, 2020

WORKING EXPERIENCE

Full-time:
China Securities Depository & Clearing Corporation, Ltd, Shanghai branch. 2002-2014
Part-time:
Xiamen University CPA Co., Ltd 1997-2001
Xiamen Capital Operation and Consulting Co., Ltd 1999-2000
Xiamen University 1998-1999

TECHNICAL SKILLS

Natural Language Processing
Create/optimize models with (un)supervised machine/deep learning methods
Financial analysis
Business data analysis
Coding with Python and R