Fake Review Detection on Yelp

Zehui Wang (wzehui), Yuzhu Zhang (arielzyz), Tianpei Qian (tianpei)

Abstract

Online reviews have become an important factor when people make purchase and business decisions. The increasing popularity of online reviews also stimulates the business of fake review writing, which refers to paid human writers producing deceptive reviews to influence readers' opinions. Our project tackles this problem by building a classifier that takes the review text and the basic information of its reviewer as input and outputs whether the review is reliable. The learning algorithms we experimented include logistic regression, linear discriminant analysis, multinomial Naive Bayes, support vector machines and neural networks. The results show that the neural network performs the best with a detection accuracy of 81.92%.

Introduction

Current research has found that the reliability of online review is in question. For example, around 20% reviews on Yelp are estimated to be faked by paid human writers [1]. In fact, the menace has soared to such serious levels that Yelp.com has launched a sting operation to publicly shame businesses who buy fake reviews.

In order to provide users with more reliable review information, we aim to build a classification system to detect fake reviews. The input to our algorithm is a review and the related information of the reviewer. We then use neural networks to output whether the review is fake or not.

Related work

The current approaches to the detection of the spam mainly focus on supervised learning using linguistic features and user-behavior features [2]. Linguistic features include word unigrams and bigrams, LIWC features and POS features [3]; user-behavior features include average review length, standard deviation in ratings and so on [4]. In our work, we use most of these common features. Instead of using unigrams and bigrams features directly, however, we fit a Latent Dirichlet Allocation(LDA) [5] topic model to the grams to obtain new features.

In terms of learning models, most researches apply logistic regression and SVM. Recently, efforts have also been made in applying deep learning models [6]. In our project, we experiment all these models and compare their performance.

Dataset and features

Dataset

The dataset is collected from Yelp.com and firstly used by Rayana and Akoglu [7] and it includes product and user information, timestamp, ratings, and a plaintext review.

The original dataset has great skew: the number of truthful reviews is larger than that of fake reviews. In our project, we randomly choose equal-sized fake and non-fake reviews from the dataset. We use a total of 16282 reviews and split it into 0.7 training set, 0.2 dev set, and 0.1 test set.

Features

Extracting predictive features from reviews and the corresponding reviewer information is the most challenging part of this project. Basically, we extract two types of features: review-centric features and reviewer-centric features.

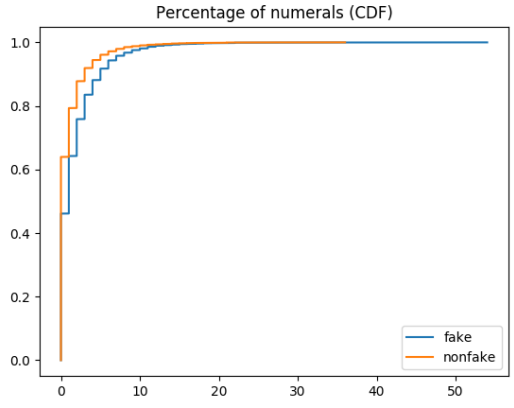
Review-centric features

1. Structural features (6): length of the review, average word length, number of sentences, average sentence length, percentage of numerals, percentage of capitalized words.
2. POS percentages (36)
3. Semantic features (2): We calculate the percentages of positive and negative opinion-bearing words in each review.
4. Unigram features (100): We extract 100 unigram features from the reviews. More about feature selection in later parts.
5. Bigram features (100): We extract 100 bigram features.

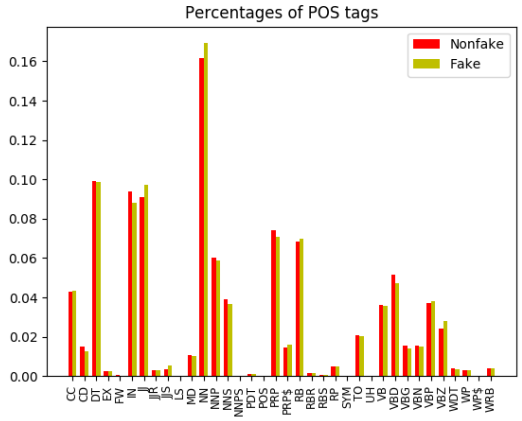
Reviewer-centric features

1. Maximum number of reviews in a day (1)
2. Percentage of reviews with positive / negative ratings (2)
3. Average review length (1)
4. Standard deviation of ratings of the reviewer's reviews (1)

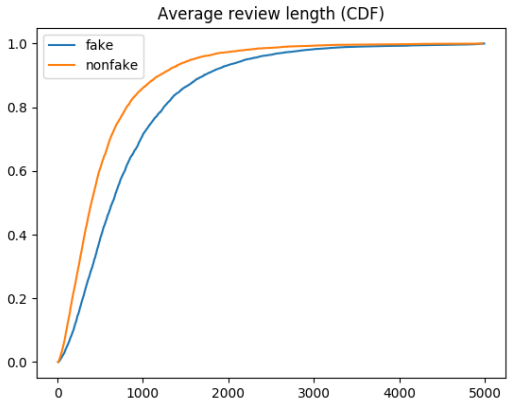
Let us inspect the distributions of these features. Below are the empirical CDFs for 4 features. We have also made similar plots for other features, but we only include 4 of them for illustration.



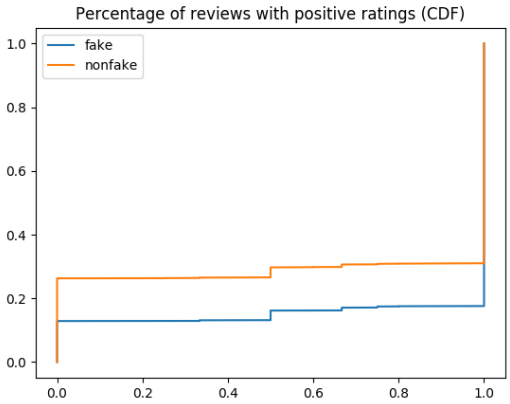
(a) Percentages of numerals



(b) Percentages of POS tags



(c) Average review length



(d) Percentages of reviews with positive ratings

We can observe that fake reviews tend to have more numerals and longer length. Also, spammers tend to give more positive ratings. Nonetheless, there is no significant difference in the percentages of POS tags.

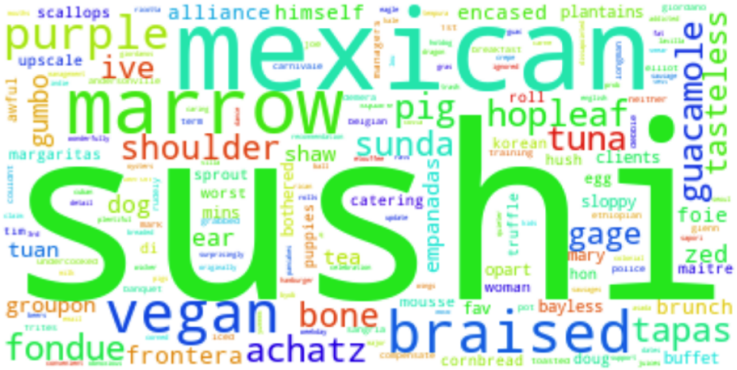
Methods

In this section, we cover how we perform feature engineering on unigrams and bigrams. Also, we give a brief summary of all the learning algorithms we have tried to build the classifier.

Feature engineering

Considering the huge feature space of unigrams and bigrams and the relative small size of our dataset, we decided to perform feature engineering on those features.

Firstly, we count the percentages of each unigram and bigram tokens for fake and non-fake reviews. We then take out the top 100 unigrams and bigrams that have the most different percentages (in terms of ratios) in fake and non-fake reviews. The results for unigrams are illustrated in the following word cloud.



We also considered a second approach — fitting LDA [5] topic models and treating the topic distribution vectors as features. The model assumes that each review is featured by a distribution over a mixture of various topics. To generate each word in the review, we first generate a topic from the topic distribution. The word is then generated from a multinomial distribution given the topic.

In the experiment part, we use the second approach to process unigrams and bigrams since it leads to better performance. This is reasonable because the second approach essentially extracts information from all unigrams and bigrams while the first approach completely omits information from unigrams and bigrams other than the 100 ones selected.

Learning algorithms

We have tried the following learning algorithms and regularization terms are added to overcome the overfitting problems.

1. Logistic Regression: Minimize logistic loss. Have the property of being well-calibrated.
2. Linear Discriminant Analysis: Generative model; The label is generated from a Bernoulli distribution. Features are normally distributed given the label.
3. Multinomial Naive Bayes: Generative model; The label is generated from a Bernoulli distribution. Given the label, the frequency of each word in the dictionary is generated from a multinomial distribution (n = length of text).
4. Support Vector Machines: Large-margin classifier. Minimize hinge loss.

5. Neural Networks: 3 hidden layers with ReLU activation; sigmoid activation for the output layer. The tuned number of neurons in the three hidden layers are 100,100,40 respectively.

Results and Discussion

Experimental results

To face the large variance of the data, all the features are normalized between zero and one. The following results are obtained by 5-fold cross validation. We can find that the Neural Network classifier has the best performance in terms of accuracy (81.92%), AUC(82.49%) and F1 score(81.42%). Other classifiers also achieve acceptable performance, which illustrates that the selected features are predictive for classifying fake and non-fake reviews.

Models	Accuracy		Area Under Curve		F1 Score	
	Dev	Test	Dev	Test	Dev	Test
LogicReg	0.7753	0.7618	0.8304	0.7753	0.7727	0.7616
LDA	0.7771	0.7649	0.8371	0.7771	0.7749	0.7666
MNB	0.7385	0.7241	0.7969	0.7395	0.7237	0.7103
SVM	0.7127	0.6924	0.7728	0.7170	0.6838	0.6544
NN (best)	0.8244	**0.8192**	0.8891	**0.8249**	0.8250	**0.8142**

Ablative analysis

As shown in the below table, the use of unigrams and bigrams alone can already achieve a satisfied results. It is not surprising since they are the majority of the feature space. Structural features and reviewer-centric features can contribute a little to the model improvement.

Features	Accuracy	AUC	F1 score
Reviewer + Structural + Unigram + Bigram	0.8192	0.8249	0.8142
Structural + Unigram + Bigram	0.8166	0.8195	0.8137
Unigram + Bigram	0.8011	0.8201	0.8037

Discussion and future work

Contrary to our expectation, adding reviewer-centric features does not boost the model performance by a significant amount. We believe it is because most reviewers only have a single review record, which makes most of our reviewer-centric features poorly defined (e.g. maximum number of reviews in a day, standard deviation of ratings). For the future, we hope to deal with a larger dataset with more reviews attributed to each reviewer, which would make these reviewer-centric features more powerful. Moreover, many other reviewer-centric feature, including geo-locations and IP addresses, can also be incorporated in our model.

Contributions

Zehui and Tianpei were mainly responsible for feature extraction. Yuzhu was mainly responsible for model fitting and testing.

References

[1] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y Zhao. Automated crowdturfing attacks and defenses in online review systems. *arXiv preprint arXiv:1708.08151*, 2017.

[2] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230. ACM, 2008.

[3] Myle Ott, Claire Cardie, and Jeffrey T Hancock. Negative deceptive opinion spam. In *Proceedings of NAACL-HLT*, pages 497–501, 2013.

[4] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[6] Yafeng Ren and Donghong Ji. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385-386:213–224, 2017.

[7] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection:bridging review networks and metadata. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 985–994, 2015.

[8] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.