Fake Review Detection Focusing on Emotional Expressions and Extreme Rating

Tavanleuang VANTA　Masaki Aono
Toyohashi University of Technology　Department of Computer Science and Engineering
vanta@kde.cs.tut.ac.jp　aono@tut.ac.jp

1. Introduction

Inexperienced buyers/customers tend to refer to historical reviews of a product/service to help them make decision. Booking a hotel room, deciding to go to a new restaurant, or picking a movie to watch, makes online reviews useful. However, the usefulness of reviews is hampered by fake reviews which are often posted by sponsored reviewers. Fake reviews can be hard to spot by human eyes and the quantity and quality of them can impact business, thus, research of fake review detection has been conducted in recent years using machine learning approach.

In this paper, we extract features from review data using only review texts and their ratings. We create some new useful features while adopt some useful literature ones, and employ them as inputs in our proposed system for fake review detection.

2. Related Work

Jindal and Liu [1] crawled non-labeled review data from Amazon.com and manually labeled them based on linguistic, behavioral, and relationship among reviews, reviewers, and products. Ott et al. [2] collected data composed of true reviews from TripAdvisor (20 most popular hotels) and fake (positive) reviews of the same hotels from Amazon Mechanical Turk (AMT). The authors defined the features such as part-of-speech (POS), unigrams, bigrams, trigrams, and psychological cues, and used the features for SVM and NB classifiers, demonstrating that the classifiers outperformed human judges.

Mukherjee et al. [3] later found substantial differences between fake reviews submitted to a review website and those artificially generated by AMT. The author used SVM trained on AMT to classify Yelp data, a commercial platform that filters suspicious reviews, and yielded maximum accuracy of 54%. Mukherjee et al. and Li et al. [4] affirm that AMT generated data cannot be representative of all types of real-life fake reviews.

Zhang et al. [5] studied both verbal and nonverbal behaviors to identify which contributed to the most to fake review detection using machine learning algorithms including SVM, NB, RF, and Decision Tree. The authors use down-sampling to extract a balanced data set from the original Yelp data set collected by Mukherjee et al. and extract 21 verbal and 26 non-verbal behavioral features in both hotel and restaurant reviews. Linguistic features such as n-grams, POS, and other non-behavioral features were not used in their study.

Ren and Zhang [6] proposed a neural network composed of Convolutional Neural Network (CNN) connected to a bi-directional Gated Recurrent Neural Network (GRNN). Ren and Ji's [7] subsequent work indicated that the best model had the review representation with the POS, n-grams, and psychological features (LIWC)

Li et al. [8] created a deep learning model named Sentence Weighted Neural Network (SWNN) to detect fake reviews in hotel, restaurant, and doctor domains. On a mixed-domain setting (all three domains) the best performance was achieved by SVM with unigram, POS, and LIWC features.

It is important to note that most existing researches exploit features from all types of data including not only review texts, but also so-called "metadata" such as user data, product data, and business data.

3. Dataset

The dataset used in this paper is restaurant reviews that were collected by A. Mukherjee, et al. [3] from Yelp. This restaurant reviews dataset consists of two sets, YelpNYC and YelpZip. The smaller set (YelpNYC) covers only reviews of the restaurants located in NYC, while the larger set (YelpZip) covers reviews collected for the restaurants based on Zipcodes of continuous region of the US map, including NJ, VT, CT, and PA. YelpNYC contains 359,052 reviews, while only 10.27% of them are fake reviews. We created a balanced review data in terms of the number of fake/true reviews from YelpNYC and use them in our experiment.

Yelp has a filtering algorithm in place that identifies fake/suspicious reviews and separates them into a filtered list. The filtered reviews are also made public; the Yelp page of a business shows the recommended reviews, while it is also possible to view the filtered/unrecommended reviews through a link at the bottom of the page [9].

We shuffled the 73,770 reviews to avoid poorly-ordered fake/true reviews before splitting them to training set (51,639 reviews) and test set (22,131 reviews)

4. Proposed Method

Our proposed method to detect fake reviews from text data can be divided into feature selection part and methodology part.

4.1. Feature Selection

All of the custom features extracted from review texts are listed in Table 1.

Table 1 List of features

1	rating
2	char count
3	word count
4	numeral token count
5	punctuation count
6	sentence count
7	title word count
8	uppercase word count
9	word density
10	average sentence length
11	noun count
12	verb count
13	adjective count
14	adverb count
15	pronoun count
16	ratio of nouns
17	ratio of verbs
18	ratio of adjectives
19	ratio of adverbs
20	ratio of pronouns
21	ratio of uppercase words
22	ratio of numeral words
23	ratio of positive words
24	ratio of negative words
25	extremity of rating

Note that 'rating' represents the rating of the restaurant given by the reviewer and 'word density' is the average length of the words used in each review.

We obtain 'rating' of each review directly from the dataset. For the second to eighth features in Table 2, the total number of characters, words, numeral tokens, punctuation, sentence, title words, uppercase words, in each review are counted respectively for 'char count', 'word count', 'numeral token count', 'punctuation count', 'sentence count', 'title word count', 'uppercase word count'.

'word density' is calculated by number of characters divided by number of words in each review. We set the calculation formula for 'word density' as (char count)/(word count +1) to avoid the division by zsero since some reviews might not have punctuations (e.g. "This place is great"), and will have the number of sentence as "0".

A python library TextBlob, which allows us to perform natural language processing (NLP) for part-of-speech task, is used to extract linguistic characteristics of fake and true reviews such as 'noun count', 'verb count', 'adjective count', 'adverb count', 'pronoun count'. Ratio of these values are themselves divided by 'word count'.

'ratio of positive words' and 'ratio of negative words' is calculated by the number of positive/negative words divided by number of words in each review, positive words and negative words are considered based on Opinion Lexicon [10].

Reviews which have rating of 1, 4, or 5 are considered as extreme rating and will be assigned a value of '1', For reviews which have rating of 2 and 3, '0' will be assigned.

4.2. Methodology

To accomplish our goal, we constructed the system as shown in Figure 1, where P1, P2, and P3 refer to the prediction results ('true' or 'fake') from SVM, MLP, and CNN+LSTM respectively. P4 refers to the prediction obtained by using majority voting method over prediction results of SVM, MLP, and CNN+LSTM.

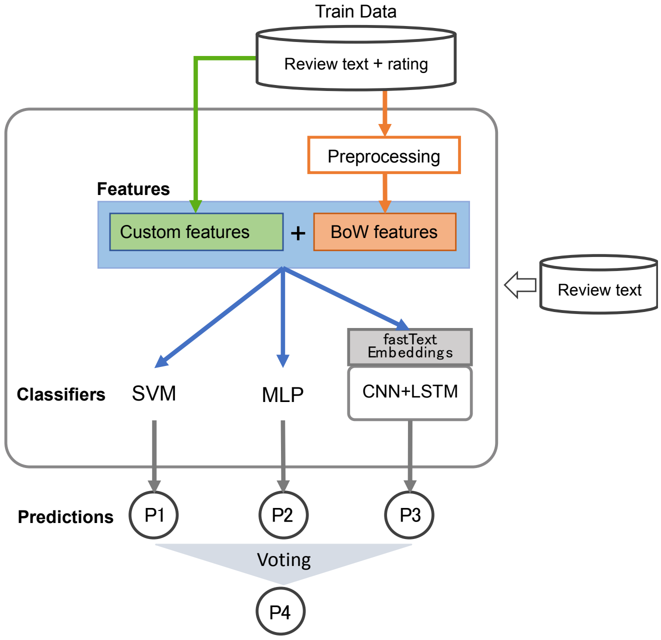

Figure 1 − Proposed system for fake review detection

4.2.1. Feature Extraction and Data Preprocessing

We employ the method mentioned in 3.1 to extract useful features from the review text and build feature vector for each review. Feature scaling is applied to standardize the range of features of data.

Customer reviews usually contain unimportant words, for instance, stem words (‘the’, ‘to’, ‘on’, etc) and time indicators (’00:30’, ‘12’, etc) which occur frequently across reviews. We perform the preprocessing tasks including lowering capital letters, removing punctuations, removing stop-words, and lemmatizing words to clean the review texts.

4.2.2. Bag-of-Words Features

Each unique word in the preprocessed review text is considered as a feature. After a dictionary of all words are created, a word-review matrix is constructed. Words that appear too infrequently in the reviews are likely to be misspells which are not useful and can introduce noise to the models. Thus, we ignore terms that appear in less than three reviews. Furthermore, words like “and” or “the” appear frequently in all reviews, therefore we apply Term Frequency-Inverse Document Frequency (TF-IDF) weighing technique to the matrix. After testing with different numbers of features, we found that using 20,000 features give the most effective for our dataset.

4.2.3. Classification Models

In our experiment, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) model, and Long Short-Term Memory (LSTM) model (with a 1D Convolutional Neural Network (CNN) layer) are used.

We choose linear SVM because it has been proved to be a powerful classifier and is suited for 2-class problem. We concatenate the feature matrix of 4.2.1 with BoW features as described here before feeding them to SVM as input.

The second model is the MLP neural network. This network consists of two or more fully-connected neural networks, which are supposed to take care of the correlation between all the inputs features during the training stage.

The third model embedded in our system is recurrent neural network LSTM which is known to perform well on text data. The model accepts the “fastText” word embeddings, whose outputs are fed into the subsequent 1D convolutional neural network (CNN) which should take care of the locally frequent patterns, followed by the LSTM.

4.2.4. Majority Voting

Here we use the simplest case of majority voting which is hard voting. We predict the class label $\hat{y}$ via majority voting of each classifier C_j:

$$\hat{y} = mode\{C_2(X), C_2(X), C_3(X)\}$$

Assuming each of our classifier give us the prediction results of a review as follows:

classifier 1 → Fake (0)
classifier 2 → True (1)
classifier 3 → True (1)

$$\hat{y} = mode\{0, 1, 1\} = 1$$

Via majority vote, we can obtain another prediction result which is ‘1’, this fourth prediction result can be seen as the output of ensemble vote classifier, provided that three of the classifiers give high and high accuracy. We will need to evaluate this method to see whether it will improve or worsen the overall result.

5. Evaluation

In this section, we describe the evaluation criteria, classification results, and the feature importance.

5.1. Evaluation Criteria

Analogous to Ott et al. [2], we choose Accuracy, Precision, Recall, and F1 Score as evaluation criteria, which is defined in (1), (2), (3), and (4) respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\,Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

The larger these values, the better the classifier is. TP (True Positive) refers to the number of positive tuples classified correctly as positive by the classifier; TN (True Negative) refers to the number of negative tuples classified correctly as negative by the classifier; FP (False Positive) refers to the number of negative tuples wrongly labeled as positive; and FN (False Negative) refers to the number of positive tuples wrongly labeled as negative.

5.2. Classification Result

The proposed custom (handcrafted) features used as inputs to all the classifiers as well as BoW features (20,000 unigrams) are shown in Table 2. Here, we set the “baseline” method as an approach using only BoW features with SVM classifier. The classification using our proposed custom features of SVM, MLP, and CNN+LSTM are Proposed 1, Proposed 2, and Proposed 3 respectively. The Majority Voting result of the three models is Proposed 4.

Table 2. Classification Results Across All Methods

Method	Accuracy	Precision	Recall	F1
Baseline	0.747	0.751	0.747	0.747
Proposed 1	0.773	0.775	0.773	0.773
Proposed 2	0.775	0.776	0.775	0.774
Proposed 3	**0.784**	**0.784**	**0.784**	**0.784**
Proposed 4	0.783	0.784	0.783	0.783

The overall results including accuracy, precision, recall, and F1 score improved by 2.2-2.6% with our proposed features compared to the baseline approach. Moreover, the score of MLP and CNN+LSTM models indicate a better performance where CNN+LSTM yields the best result. On the other hand, the predictions from voting of the three classifiers give a slightly lower score than of the CNN+LSTM model.

5.3. Feature Importance

The effectiveness of 4.2.1 and 4.2.3 features on fake review detection performance can be simulated by Random Forest model, which is renowned for easily estimating the importance among features. The top 20 most effective features and their relative importance values are shown in Figure 4. Note that BoW features are written within single quotes.

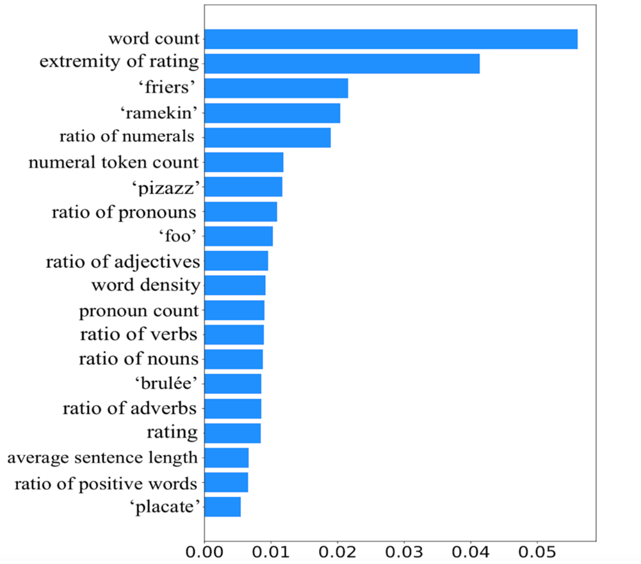

Figure 4 – Top 20 effective features

6. Conclusion

In this paper we investigated the effects of applying many useful features which were extracted from review texts and ratings to be used in SVM, MLP, and CNN+LSTM. A real-life dataset of 73,770 reviews and their ratings was used to train and test the models. Accuracy, Precision, Recall, and F1 Score of evaluation measures were used to verify the experimental results. The classification results showed that useful custom features extracted from only review texts can boost the models' performance compared to using BoW features alone. In the future, we hope to experiment with a larger dataset as well as the dataset of other business to see how the performance of the system will change. Furthermore, we would like to add metadata such as reviewer-centric features obtained from user data− maximum reviews in a day, percentage of positive/negative reviews, and similarity of the user's reviews, IP address; and product/service data which have been used in related to boost the performance and accuracy.

Acknowledgment

A part of this research was carried out with the support of the Grant-in-Aid for Scientific Research (B) (issue number 17H01746).

References

[1] Jindal, N. & Liu, B. (2008), Opinion spam and analysis, in 'Proceedings of the 2008 International Conference on Web Search and Data Mining'.

[2] Ott, M., Choi, Y., Cardie, C. & Hancock, J. T. (2011), 'Finding deceptive opinion spam by any stretch of the imagination'.

[3] Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. (2013), 'What yelp fake review filter might be doing?'.

[4] Li, J., Ott, M., Cardie, C. & Hovy, E. (2014), Towards a general rule for identifying deceptive opinion spam, in 'Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Baltimore, Maryland, pp. 1566–1576.

[5] Zhang, D., Zhou, L., Kehoe, J. L. & Kilic, I. Y. (2016), 'What online reviewer behaviors really matter? effects of verbal and nonverbal behaviors on detection of fake online reviews'.

[6] Ren, Y. & Zhang, Y. (2016), Deceptive opinion spam detection using neural network, in 'Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers'.

[7] Ren, Y. & Ji, D. (2017), 'Neural networks for deceptive opinion spam detection'.

[8] Li, L., Qin, B., Ren, W. & Liu, T. (2017), 'Document representation and feature combination for deceptive spam review detection', Neurocomputing 254, 33–41.

[9] Shebuti Rayana and Leman Akoglu (2016),' Collective Opinion Spam Detection using Active Inference'.

[10] Bing Liu, Minqing Hu and Junsheng Cheng. 'Opinion Observer: Analyzing and Comparing Opinions on the Web.', Proceedings of the 14th international World Wide Web conference (WWW-2005), May 10-14, 2005, Chiba, Japan.