Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews

Arjun Mukherjee[†], Vivek Venkataraman[†], Bing Liu[†], Natalie Glance[‡]
[†]University of Illinois at Chicago, [‡]Google Inc.
arjun4787@gmail.com, vivek1186@gmail.com, liub@cs.uic.edu, nglance@google.com

ABSTRACT

In recent years, fake review detection has attracted significant attention from both businesses and the research community. For reviews to reflect genuine user experiences and opinions, detecting fake reviews is an important problem. Supervised learning has been one of the main approaches for solving the problem. However, obtaining labeled fake reviews for training is difficult because it is very hard if not impossible to reliably label fake reviews manually. Existing research has used several types of pseudo fake reviews for training. Perhaps, the most interesting type is the pseudo fake reviews generated using the Amazon Mechanical Turk (AMT) crowdsourcing tool. Using AMT crafted fake reviews, [36] reported an accuracy of 89.6% using only word *n*-gram features. This high accuracy is quite surprising and very encouraging. However, although fake, the AMT generated reviews are not *real* fake reviews on a commercial website. The Turkers (AMT authors) are not likely to have the same psychological state of mind while writing such reviews as that of the authors of real fake reviews who have real businesses to promote or to demote. Our experiments attest this hypothesis. Next, it is naturally interesting to compare fake review detection accuracies on pseudo AMT data and real-life data to see whether different states of mind can result in different writings and consequently different classification accuracies. For real review data, we use filtered (fake) and unfiltered (non-fake) reviews from Yelp.com (which are closest to ground truth labels) to perform a comprehensive set of classification experiments also employing only *n*-gram features. We find that fake review detection on Yelp's real-life data only gives 67.8% accuracy, but this accuracy still indicates that *n*-gram features are indeed useful. We then propose a novel and principled method to discover the precise difference between the two types of review data using the information theoretic measure KL-divergence and its asymmetric property. This reveals some very interesting psycholinguistic phenomena about forced and natural fake reviewers. To improve classification on the real Yelp review data, we propose an additional set of behavioral features about reviewers and their reviews for learning, which dramatically improves the classification result on real-life opinion spam data.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis; J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Experimentation, Measurement

Keywords

Opinion spam, Fake review detection, Behavioral analysis

1. INTRODUCTION

Online reviews are increasingly used by individuals and organizations to make purchase and business decisions. Positive reviews can render significant financial gains and fame for businesses and individuals. Unfortunately, this gives strong incentives for imposters to game the system by posting fake reviews to promote or to discredit some target products or businesses. Such individuals are called *opinion spammers* and their activities are called *opinion spamming*. In the past few years, the problem of spam or fake reviews has become widespread, and many high-profile cases have been reported in the news [44, 48]. Consumer sites have even put together many clues for people to manually spot fake reviews [38]. There have also been media investigations where fake reviewers blatantly admit to have been paid to write fake reviews [19]. The analysis in [34] reports that many businesses have tuned into paying positive reviews with cash, coupons, and promotions to increase sales. In fact the menace created by rampant posting of fake reviews have soared to such serious levels that Yelp.com has launched a "sting" operation to publicly shame businesses who buy fake reviews [43].

Since it was first studied in [11], there have been various extensions for detecting individual [25] and group [32] spammers, and for time-series [52] and distributional [9] analysis. The main detection technique has been supervised learning. Unfortunately, due to the lack of reliable or gold-standard fake review data, existing works have relied mostly on ad-hoc fake and non-fake labels for model building. In [11], supervised learning was used with a set of review centric features (e.g., unigrams and review length) and reviewer and product centric features (e.g., average rating, sales rank, etc.) to detect fake reviews. Duplicate and near duplicate reviews were assumed to be fake reviews in training. An AUC (*Area Under the ROC Curve*) of 0.78 was reported using logistic regression. The assumption, however, is too restricted for detecting generic fake reviews. The work in [24] used similar features but applied a co-training method on a manually labeled dataset of fake and non-fake reviews attaining an F1-score of 0.63. The result too may not be completely reliable due to the noise induced by human labels in the dataset. Accuracy of human labeling of fake reviews has been shown to be quite poor [36].

Another interesting thread of research [36] used Amazon Mechanical Turk (AMT) to manufacture (by crowdsourcing) fake hotel reviews by paying (US$1 per review) anonymous online workers (called *Turkers*) to write fake reviews by portraying a hotel in a positive light. 400 fake positive reviews were crafted using AMT on 20 popular Chicago hotels. 400 positive reviews from Tripadvisor.com on the same 20 Chicago hotels were used as non-fake reviews. The authors in [36] reported an accuracy of 89.6% using only word bigram features. Further, [8] used some deep syntax rule based features to boost the accuracy to 91.2%.

The significance of the result in [36] is that it achieved a very high accuracy using only word *n*-gram features, which is both very surprising and also encouraging. It reflects that while writing fake reviews, people do exhibit some linguistic differences from other genuine reviewers. The result was also widely reported in the news, e.g., The New York Times [45]. However, a weakness of this study is its data. Although the reviews crafted using AMT are fake, they are not *real* "fake reviews" on a commercial website. The Turkers are not likely to have the same psychological state of mind when they write fake reviews as that of authors of real fake reviews who have real business interests to promote or to demote. If a real fake reviewer is a business owner, he/she knows the business very well and is able to write with sufficient details,

Technical Report, Department of Computer Science (UIC-CS-2013-03). University of Illinois at Chicago.

rather than just giving glowing praises of the business. He/she will also be very careful in writing to ensure that the review sounds genuine and is not easily spotted as fake by readers. If the real fake reviewer is paid to write, the situation is similar although he/she may not know the business very well, this may be compensated by his/her experiences in writing fake reviews. In both cases, he/she has strong financial interests in the product or business. However, for an anonymous Turker, he/she is unlikely to know the business well and does not need to write carefully to avoid being detected because the data was generated for research, and each Turker was only paid US$1 for writing a review. This means that his/her psychological state of mind while writing can be quite different from that of a real fake reviewer. Consequently, their writings may be very different, which is indeed the case as we will see in § 2, 3.

To obtain an in-depth understanding of the underlying phenomenon of opinion spamming and the hardness for its detection, it is scientifically very interesting from both the fake review detection point of view and the psycholinguistic point of view to perform a comparative evaluation of the classification results of the AMT dataset and a real-life dataset to assess the difference. This is the first part of our work. Fortunately, Yelp.com has excellent data for this experiment. Yelp.com is one of the largest hosting sites of business reviews in the United States. It filters reviews it believes to be suspicious. We crawled its filtered (fake) and unfiltered (non-fake) reviews. Although, the Yelp data may not be perfect, its filtered and unfiltered reviews are likely to be the closest to the ground truth of real fake and non-fake reviews since Yelp engineers have worked on the problem and been improving their algorithms for years. They started to work on filtering shortly after their launch in 2004 [46]. Yelp is also confident enough to make its filtered and unfiltered reviews known to the public on its Web site. We will further discuss the quality of Yelp's filtering and its impact on our analysis in § 7.

Using exactly the same experiment setting as in [36], the real Yelp data only gives 67.8% accuracy. This shows that (1) *n*-gram features are indeed useful and (2) fake review detection in the real-life setting is considerably harder than in the AMT data setting in [36] which yielded about 90% accuracy. Note that a balanced data (50% fake and 50% non-fake reviews) was used as in [36]. Thus, by chance, the accuracy should be 50%. Results in the natural distribution of fake and non-fake will be given in § 2.

An interesting and intriguing question is: What exactly is the difference between the AMT fake reviews and Yelp fake reviews, and how can we find and characterize the difference? This is the second part of our work. We propose a novel and principled method based on the information theory measure, KL-divergence and its asymmetric property. Something very interesting is found.

1. The word distributions of fake reviews generated using AMT and non-fake reviews from Tripadvisor are widely different, meaning a large number of words in the two sets have very different frequency distributions. That is, the Turkers tend to use different words from those of genuine reviewers. This may be because the Turkers did not know the hotels well and/or they did not put their hearts into writing the fake reviews. That is, the Turkers did not do a good job at "faking". This explains why the AMT generated fake reviews are easy to classify.
2. However, for the real Yelp data, the frequency distributions of a large majority of words in both fake and non-fake reviews are very similar. This means that the fake reviewers on Yelp have done a good job at faking because they used similar words as those genuine (non-fake) reviewers in order to make their reviews sound convincing. However, the asymmetry of KL-divergence shows that certain words in fake reviews have much higher frequencies than in non-fake reviews. As we will see in § 3, those high frequency words actually imply pretense and deception. This indicates that Yelp fake reviewers have *overdone* it in making their reviews sound genuine as it has left footprints of linguistic pretense. The combination of the two findings explains why the accuracy is better than 50% (random) but much lower than that of the AMT data set.

The next interesting question is: Is it possible to improve the classification accuracy on the real-life Yelp data? The answer is yes. We then propose a set of behavioral features of reviewers and their reviews. This gives us a large margin improvement as we will see in § 6. What is very interesting is that using only the new behavioral features alone does significantly better than bigrams used in [36]. Adding bigrams only improve performance slightly.

To conclude this section, we also note the other related works on opinion spam detection. In [12], different reviewing patterns are discovered by mining unexpected class association rules. In [25], some behavioral patterns were designed to rank reviews. In [49], a graph-based method for finding fake store reviewers was proposed. None of these methods perform classification of fake and non-fake reviews which is the focus of this work. Several researchers also investigated review quality [e.g., 26, 54] and helpfulness [17, 30]. However, these works are not concerned with spamming. A study of bias, controversy and summarization of research paper reviews was reported in [22, 23]. This is a different problem as research paper reviews do not (at least not obviously) involve faking. In a wide field, the most investigated spam activities have been Web spam [1, 3, 5, 35, 39, 41, 42, 52, 53, 55] and email spam [4]. Recent studies on spam also extended to blogs [18, 29], online tagging [20], clickbots [16], and social networks [13]. However, the dynamics of all these forms of spamming are quite different from those of opinion spamming in reviews.

We now summarize the main results/contributions of this paper:

1. It performs a comprehensive set of experiments to compare classification results of the AMT data and the real-life Yelp data. The results show that classification of the real-life data is considerably harder than classification of the AMT pseudo fake reviews data generated using crowdsourcing [36]. Furthermore, our results show that models trained using AMT fake reviews are not effective in detecting real fake reviews as they are not representative of real fake reviews on commercial websites.
2. It proposes a novel and principled method to find the precise difference between the AMT data and the real-life data, which explains why the AMT data is much easier to classify. Also importantly, this enables us to understand the psycholinguistic differences between real fake reviewers who have real business interests and cheaply paid AMT *Turkers* hired for research [36]. To the best of our knowledge, this has not been done before.
3. A set of behavioral features is proposed to work together with *n*-gram features. They improve the detection accuracy dramatically. Interestingly, we also find that behavioral features alone already can do significantly better than *n*-grams. Again, this has not been reported before. We also note that the AMT generated fake reviews do not have the behavior information of reviewers like that on a real website, which is also drawback.

2. A COMPARATIVE STUDY

This section reports a comprehensive set of classification experiments using the real-life data from Yelp and the AMT data from [36]. We will see a large difference in accuracy between the two datasets. In Section 3 we will characterize the difference.

2.1 The Yelp Review Dataset

As described earlier, we use reviews from Yelp.com. To ensure high credibility of user opinions posted on Yelp, it uses a filtering algorithm to filter suspicious reviews and prevents them from showing up on the businesses' pages. Yelp, however, does not delete those filtered reviews but puts them in a filtered list, which is publicly available. According to CEO, Jeremy Stoppelman,

Yelp's mission is to provide users with the most trustworthy content. It is achieved through its automated review filtering process [46]. He stated that the review filter has "evolved over the years; it's an algorithm our engineers are constantly working on." [46]. Yelp purposely does not reveal the clues that go into its filtering algorithm as doing so can lessen the filter's effectiveness [27]. Although Yelp's review filter has been claimed to be highly accurate by a study in BusinessWeek [51], Yelp accepts that the filter may catch some false positives [10], and is ready to accept the cost of filtering a few legitimate reviews than the infinitely high cost of not having an algorithm at all which would render it become a *lassez-faire* review site that people stop using [27].

It follows that we can regard the filtered and unfiltered reviews of Yelp as sufficiently reliable and possibly closest to the ground truth labels (fake and non-fake) available in the real-life setting. To attest this hypothesis, we further analyzed the quality of the Yelp dataset (see § 5.2) where we show that filtered reviews are strongly correlated with abnormal spamming behaviors (see § 7 as well). The correlation being statistically significant (p<0.01) renders high confidence on the quality of labels in the dataset.

In this work, we use filtered (fake) and unfiltered (non-fake) reviews from Yelp.com across 85 hotels and 130 restaurants in the Chicago area. To avoid any bias we consider a mixture of popular and unpopular hotels and restaurants (based on the number of reviews) in the dataset. Table 1 gives the dataset statistics. We note that fake and non-fake distribution is skewed or imbalanced.

Note that in [36], the classification was performed on 400 fake reviews from AMT and 400 reviews from Tripadvisor which were assumed to be non-fake. This 50% *class distribution* is called *balanced data*. However, in real life, this is not the case as shown in Table 1 because in practice the proportion of fake is much smaller than non-fake reviews. For example, Yelp filters about 14% of reviews as suspicious. Thus, the natural distribution on Yelp is about 14% fake and 86% non-fake, which indicates a *skewed* (*class*) *distribution* or *imbalanced data*. The skewed distribution can have a major negative impact on the classification accuracy. We perform two kinds of experiments using balanced data (to compare with the existing work in [36]) and imbalanced data following the natural distribution of the two classes (fake and non-fake) to assess the real accuracy.

2.2 Yelp Data Classification Experiments

We now report the classification results using the real-life Yelp data under both balanced and natural class distribution settings.

2.2.1 Data Preparation

All our experiments are based on 5-fold *C*ross *V*alidation (CV), which was also done in [36]. The training and test data are also prepared accordingly. It is well known that highly imbalanced training data often produce poor models [2, 6]. To build a good model for imbalanced data, one of the common techniques used in machine learning is to employ fewer frequent class instances to make the training data more balanced [47]. In our experiments, we vary the proportions of fake and non-fake (frequent class) reviews.

Training data of each fold: Let r be the percent of fake reviews in the data. r varies from the natural distribution (i.e., actual proportions of fake/non-fake in the Yelp data) to 50% (i.e., balanced distribution). To produce different training sets, we use all fake reviews and vary the number of non-fake reviews (since we have many more non-fake reviews).

Test data of each fold: For the test data, we use two settings: i) balanced data, 50% fake and 50% non-fake (50:50) and ii) natural distribution (N.D.) with the same proportions of fake and non-fake reviews as in each domain in Table 1. As noted in [2, 6], correct classifier evaluation should use the natural distribution.

	fake	non-fake	% fake	Total # reviews	# reviewers
Hotel	802	4876	14.1%	5678	5124
Restaurant	8368	50149	14.3%	58517	35593

Table 1: Dataset statistics

I. Hotel Domain

F. V. A.	C.D.	P	R	F1	A
Boolean	50:50	62.9	76.6	68.9	65.6
	N.D.	20.3	76.6	31.9	57.4
TF	50:50	68.2	56.8	61.8	65.0
	N.D.	24.2	56.8	33.9	70.4
TF-IDF	50:50	73.2	37.2	48.9	61.4
	N.D.	29.8	37.2	33.1	78.2

(a): Unigrams

P	R	F1	A
61.1	79.9	69.2	64.4
19.8	79.9	31.7	54.6
67.0	53.9	59.8	63.7
23.9	53.9	33.1	72.1
73.9	34.5	46.3	60.8
31.7	34.5	32.9	80.9

(b): Bigrams

II. Restaurant Domain

F. V. A.	C.D.	P	R	F1	A
Boolean	50:50	64.3	76.3	69.7	66.9
	N.D.	20.4	76.3	32.2	60.1
TF	50:50	68.5	58.3	63.0	65.8
	N.D.	23.3	58.3	33.3	71.1
TF-IDF	50:50	69.2	57.2	62.6	65.9
	N.D.	23.9	57.2	33.7	71.9

(a): Unigrams

P	R	F1	A
64.5	79.3	71.1	67.8
20.2	79.3	32.1	58.6
69.1	55.2	61.4	66.1
22.9	55.2	32.4	72.6
72.1	52.5	60.8	64.2
26.4	52.5	35.1	76.4

(b): Bigrams

Table 2: SVM 5-fold CV results, P: Precision, R: Recall, F1: F1-Score on the fake class, A: Accuracy in %. Training uses balanced data (50:50). Testing uses two different class distributions (C.D.): 50:50 (balanced) and Natural Distribution (N.D.). Results report different Feature Value Assignment (F.V.A.) schemes for both unigram and bigram features for hotel and restaurant domains.

Features	C.D.	P	R	F1	A
POS Unigrams	50:50	56.0	69.8	62.1	57.2
	N.D.	15.5	69.8	25.4	47.6
W-Bigrams + POS-Bigrams	50:50	63.2	73.4	67.9	64.6
	N.D.	20.0	73.4	31.4	55.1
W-Bigrams + Deep Syntax	50:50	62.3	74.1	67.7	64.1
	N.D.	19.9	74.1	31.3	54.7
W-Bigrams + POS Seq. Pat.	50:50	63.4	74.5	68.5	64.5
	N.D.	20.2	74.5	31.7	55.1

(a): Hotel

P	R	F1	A
59.5	70.3	64.5	55.6
16.9	70.3	27.2	51.9
65.1	72.4	68.6	68.1
21.6	72.4	33.3	59.6
65.8	73.8	69.6	67.6
20.1	73.8	31.6	58.7
66.2	74.2	69.9	67.7
20.3	74.2	31.8	58.9

(b): Restaurant

Table 3: SVM 5-fold CV results, Training: 50:50, Testing: two different class distributions (C.D.) 50:50 and N.D. W means word unigram, W-Bigram denotes word bigrams, and POS denotes part of speech tags.

2.2.2 Results Using Balanced Training Data

Tables 2 and 3 report the classification results using classifiers learned from balanced (50:50) training data, and tested on both 50:50 and natural distribution (N.D.) data. For model building, we use SVM (the SVM^{Light} system [14]). Our experiments showed that linear kernel SVM outperformed rbf, sigmoid, and polynomial kernels. Hence, we only report results using the linear kernel, which has been shown very effective for text classification in many prior works, e.g., [15].

Results using Boolean, TF, and TF-IDF features: Table 2 shows the results using various *F*eature *V*alue *A*ssignments (F.V.A.) schemes: Boolean, TF, and TF-IDF with both unigram and bigram (bigrams are inclusive of unigrams) features. Higher order n-grams did not help. We also tried different feature selection schemes and the naïve Bayes classifier, but resulted in slightly poorer models. Hence, we omit their results. From Table 2, we make the following observations:

1. F1 results for 50:50 exceeds N.D. by a large margin across both domains and F.V.A. schemes, showing that detecting fake reviews is much harder in the natural class distribution. Note

that keeping other settings fixed the recall values for 50:50 and N.D. are same because the total numbers of fake reviews in the test data for 50:50 and N.D. are the same, and only the number of non-fake reviews varies.

2. For the balanced 50:50 test setting, Boolean F.V.A. scheme performed best for F1 and accuracy metrics. TF and TF-IDF features render slight improvements in precision but at the expense of large drops in recall resulting in lower F1.
3. For the N.D. setting, all three F.V.A. schemes give similar F1 scores with a slight edge for TF and TF-IDF. Compared to balanced data the accuracy is higher, but this improvement in accuracy in the N.D. setting is not useful due to imbalanced test data [47]. The classifier can get a high accuracy without detecting any fake reviews by merely classifying all reviews as non-fake. Thus, for imbalanced data, accuracy is not a good metric of classification performance. F1 is much better.
4. Unigrams and bigrams performed very similarly.

Overall Boolean and bigram combination is slightly better than other combinations. Hence, for the subsequent experiments, we use Boolean feature value assignment (F.V.A). We will still use unigram and bigram models for a rich comparison.

Results using POS and complex existing features: Prior works have showed that POS (part-of-speech) based features can be useful in building classifiers. The work in [31] proposed POS sequence patterns features. A *POS sequence pattern* is a sequence of POS tags that satisfy two constraints: minimum frequency/support (*minsup*) and *adherence* which are computed using symmetric conditional probability proposed in [40]. As suggested in [31], we use *minsup* = 30%, *adherence* = 20%, and mine all sequence patterns. This generates a new class of features.

In [8], a set of deep syntax features were used to improve the accuracy on the AMT balanced data [36]. These deep syntax based features are some lexicalized (e.g., PRP → "you") and un-lexicalized (e.g., NP_2 → NP_3 SBAR) production rules involving immediate or grandparent nodes based on Probabilistic Context Free Grammar (PCFG) parse trees.

We experimented with POS unigrams, POS sequence patterns, and deep syntax based features (using the Stanford Parser) on our real-life dataset. Table 3 reports the results using Boolean F.V.A. (as it performed best in Table 2). We note the following observations from Table 3:

1. Simply using POS unigrams produce poor results in both 50:50 and N.D. settings. POS unigrams are thus not discriminative.
2. Word bigrams and POS bigrams (Table 3; row 2) render slight improvements in accuracy for 50:50 setting over word bigrams in Table 2. For N.D., we, find about 6% drop in recall scores.
3. Deep syntax features (Table 3; row 3) slightly improve recall over W-bigrams + POS bigrams (Table 3, row 2) but reduces accuracy in 50:50 setting. For the N.D. setting, using deep syntax features render a small drop in F1.
4. POS sequence patterns (Table 3, row 4) perform similarly to other methods except POS unigrams (which is the worst).

Hence, we see that neither deep syntax nor POS sequence patterns are helpful for the real-life dataset[1]. POS features also make little difference compared to word unigram and bigram models (Table 2). Hence, for subsequent experiments, we only report results using unigram and bigram models with Boolean F.V.A.

2.2.3 Varying % of Fake Reviews in Training

Since we have skewed data, we can vary the percentage of fake reviews in training and see which gives the best results on the test data. As noted early, we do this by adding more non-fake reviews as it is the majority class. In Figure 1, we vary the proportion of fake reviews in the training data from 0% to 50% (50:50) in the *x*-axis. The first point on the right is the natural distribution. For testing, only the natural distribution is used, which is the realistic situation. From Figure 1, we note the following observations:

1. As the fake class proportion increases in the training set, there is a monotonic increase in recall across both unigram and bigram models in both domains. When the model is trained on the natural distribution (which is the starting point of the curves), the test results are extremely poor, almost 0 recall, which is expected as skewed data builds poor models [39].
2. For unigrams, the precision first increases up to a certain value and then decreases to roughly 20% for both domains. This shows that when the model is trained on 50:50 setting but evaluated on the natural distribution, only 20% of the predicted fake reviews are actually fake. This means that in natural distribution, detecting fake reviews is still very hard. For bigrams, the precision behaves slightly differently.
3. The F1 increases first and then stabilizes when the fake class distribution reaches 40% (at 50%, it drops very slightly).
4. Lastly, the accuracy decreases as the fake class training data increases due to the imbalanced test data, but as noted earlier, for skewed test data, accuracy is not a good measure.

In summary, we can see that when evaluated/tested in the natural class distribution, fake review detection is very hard. It only achieves 0.3-0.4 F1 scores with very low precision.

2.3 Comparison with Ott et al. [36]

The previous section conducted experiments on real-life fake review datasets. In [36], Ott et al. reported 89.6% accuracy for review spam detection using bigram features based on AMT fake reviews. Turkers (anonymous online workers) were asked to "synthesize" hotel reviews by assuming that they work for the hotel's marketing department and their boss wants them to write reviews to portray the hotel in the positive light. 400 independent Turkers wrote one such review each across 20 most popular Chicago hotels. These 400 reviews were treated as fake. The non-fake class comprised of 400 5-star reviews of the same 20 hotels from TripAdvisor.com. Since Turkers were asked to portray the hotels in positive light [36], it means that they wrote 4-5 star reviews. We thus also use 4-5 star fake reviews in our real-life Yelp data. To keep the natural distribution of fake and non-fake reviews, we also use 4-5 star non-fake reviews for model building (instead of using only 5 star reviews). Note that our experiments in § 2.2 used both positive and negative reviews.

As [36], we also use reviews of only "popular" Chicago hotels and restaurants from our dataset (see Table 1). Applying the two restrictions (popularity and 4-5 ★), we obtained 416 fake and 3090 non-fake reviews from our hotel domain (see Table 4). 416 fake reviews are quite close and comparable to the number in Ott et al., [36] who used 400. We first compare results using hotel reviews in our real-life data as [36] used only hotel reviews. Training used the 50:50 balanced setting as in [36]. Table 5 reports 5-fold cross-validation results of SVM across both 50:50 and N.D. test settings using unigrams and bigrams. We also report results of our implementation on their data for reliable comparison. Note that Ott et al. [36] also tried adding LIWC features [36] on top of bigrams but LIWC features made marginal difference (increased the accuracy only to 89.8% from 89.6%). Thus, we do not use the LIWC features. From Table 5, we note the following:

1. Our implementation on AMT data of Ott et al. [36] produces comparable results[2]. This renders confidence in our

[1] Prior work [8] reported improvements using deep syntax over bigrams on the AMT generated dataset in [36]. However, there are fundamental differences between AMT data and our real-life spam dataset as we will detail in § 3.

[2] Minor variations of classification results are common due to different binning in cross-validation, tokenization, etc. [36] did not provide some details about their experiment settings, e.g., feature value assignment, SVM kernel, etc.

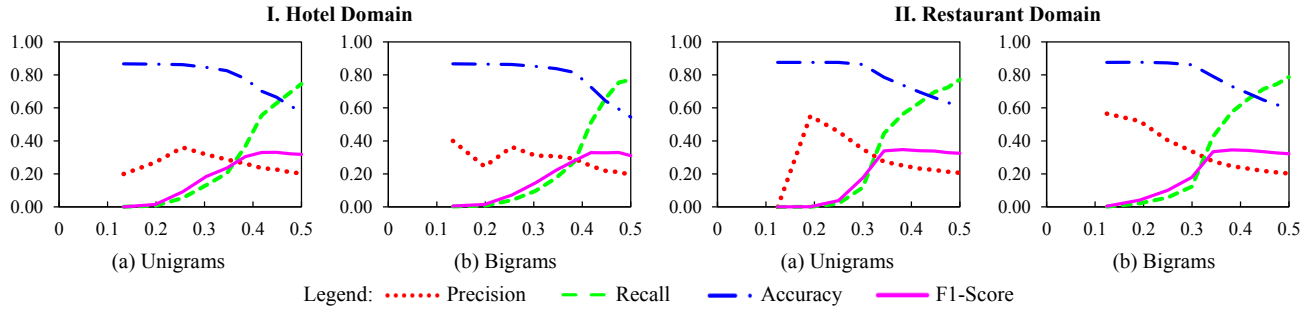


Figure 1: SVM 5-fold CV metrics by varying the proportion of fake reviews in the training data. For testing, natural distribution is used.

	Ho. 4-5 ★	Re. 4-5 ★	Re. 4-5 ★ Am.
Fake	416	5992	1511
Non-fake	3090	44238	12887

Table 4: 4-5 star reviews from Hotel and Restaurant domains. Ho: Hotel, Re: restaurant, Am: American cuisine restaurant.

n	C.D.	P	R	F1	A
Uni	50:50	86.7	89.4	88.0	87.8
	N.D.	-	-	-	-
Bi	50:50	88.9	89.9	89.3	88.8
	N.D.	-	-	-	-

(a) Ott et al. [36]

P	R	F1	A
65.1	78.1	71.0	67.6
17.7	78.1	28.8	54.3
61.1	82.4	70.2	64.9
17.0	82.4	28.2	50.8

(b) Hotel

P	R	F1	A
65.1	77.4	70.7	67.9
19.6	77.4	31.3	59.5
64.9	81.2	72.1	68.5
19.2	81.2	31.5	57.5

(c) Restaurant

P	R	F1	A
64.7	78.2	70.8	67.6
19.8	78.2	31.6	59.1
65.3	82.3	72.8	68.2
19.3	82.3	31.3	57.1

(d) American Cuisine

Table 5: Comparison with Ott et al. [36] based on SVM 5-fold CV results. Training: 50:50 balanced data, Testing: 50:50 and Natural Distribution (N.D.). Feature Sets: unigrams (Uni) and bigrams (Bi).

implementation. We cannot perform testing on the N.D. setting for Ott et al. data as it is too small (only 400 fake reviews) to give a reliable result for such an experiment setting.

2. Although Ott et al. [36] reported 89.6% accuracy, our real-life spam data from the hotel domain gave 67.6% accuracy for 50:50 setting (Table 5, b). This shows that for real-life spam data, the problem is much harder than using the AMT data.

To further confirm our results, we also report results using the restaurant domain data in Table 5 (c), which shows a similar (67.9%) accuracy for 50:50 setting which bolsters our confidence about the problem difficulty in the real-life setting.

Comparing with Table 2 (Boolean and 50:50 settings), we find that in both domains, SVM renders slightly better classification accuracy and F1 using only positively rated reviews (Table 5, b, c). For the N.D. setting, we find an improvement in recall scores over Table 2. However, there is a reduction in precision and in F1 scores for fake reviews in the N.D. setting.

We also experimented with reviews of a specific genre of restaurant, American (single) cuisine in Tables 5(d). For data refer to Table 4. We see that using only one (American) cuisine, SVM classification performs similarly to multi-cuisine restaurant data Table 5(c). The minor differences are not statistically significant.

In summary, we can say that in the real-life spam dataset and the natural setting, the problem of fake review classification is much harder than using the pseudo AMT data in [36] and in [48] (which reported around 91.2% accuracy with deep syntax features). Later, in § 4, we will see that AMT fake reviews are not so representative of real fake reviews in commercial websites. Below, we perform a deep investigation into the two data sets to find and to characterize their precise differences.

3. WORD DISTRIBUTION ANALYSIS

To understand the large disparity in classification results between the AMT data [36] and the real-life Yelp review data, we analyze the word unigram distributions as text classification relies principally on word distributions in different classes. Our results in § 2 also show that bigrams perform similarly. We thus explore the distribution of words in fake and non-fake reviews.

We used Good-Turing smoothed unigram language models[3] of fake and non-fake reviews to compute word probability distributions. As language models are constructed on documents, we construct a big document of fake reviews (respectively non-fake reviews) by merging all fake (non-fake) reviews together.

To compute the word distribution differences among fake and non-fake reviews, we use Kullback–Leibler (KL) divergence: $KL(F||N) = \sum_i F(i) \log_2 \left(\frac{F(i)}{N(i)}\right)$, where $F(i)$ and $N(i)$ are the respective probabilities of word i in fake and non-fake reviews. The motivation behind using KL-divergence is as follows: In information theory, the Kraft–McMillan theorem [21] establishes that any coding scheme (generative model) for coding (generating) a message (review, in our context) as a sequence of random variables (terms) where each variable (term) takes one value (word/token) out of a set of possibilities (vocabulary of words) can be seen as representing an implicit probability distribution (i.e., the underlying generative language model). $KL(F||N)$ is then defined as the expected extra information (linguistic distributional difference) that must be exhibited if a code (generative language model) that is optimal for (fitted against) a given (wrong/different) distribution N is used to compute messages (generate reviews using language model fitted on N) than using a code based on the true distribution F (generative language model fitted on F).

Thus, in our context of fake and non-fake reviews, $KL(F||N)$ provides a quantitative estimate of *how much* do fake reviews linguistically (according to the frequency of word usage) differ from non-fake reviews. $KL(N||F)$ can be interpreted analogously.

An important property about KL-divergence (KL-Div.) is that it is asymmetric, i.e., $KL(F||N) \neq KL(N||F)$. Its symmetric extension is the Jensen-Shannon (JS) divergence: $JS = \frac{1}{2}KL(F||M) + \frac{1}{2}KL(N||M)$, where $M = \frac{1}{2}(F + N)$. However, the asymmetry of KL-Div. can provide us some crucial information. In Table 6, we report KL, $\Delta KL = KL(F||N) - KL(N||F)$, and JS results for various datasets. We note the following interesting observations:

- For the AMT data of [36] (Table 6, row 1), we get $KL(F||N) \approx KL(N||F)$ and $\Delta KL \approx 0$. However, for our real-life spam datasets (Table 6, rows 2-5), there are major differences, $KL(F||N) > KL(N||F)$ and $\Delta KL > 1$.
- For the symmetric Jensen–Shannon (JS) divergence, the divergence of fake and non-fake word distributions in the AMT data of [36] is much larger (almost double) than our real-life Yelp data, which explains why the AMT data is much easier to classify. We will discuss this in further details below.

[3] Using Kylm - The Kyoto Language Modeling Toolkit

We now investigate the differences of KL-divergences of the two types of data, which will clearly elucidate the dissimilarity between our real-life data and the AMT data.

From the definition of KL-divergence, it implies that those words which have very high probabilities in F and very low probability in N contribute most to KL-divergence, $KL(F||N)$. To examine the word-wise contribution to ΔKL, we compute the word KL-divergence difference for each word, ΔKL^i_{Word} as follows:

$$\Delta KL^i_{Word} = KL_{Word}(F_i||N_i) - KL_{Word}(N_i||F_i),$$

where $KL_{Word}(F_i||N_i) = F(i)\log_2\left(\frac{F(i)}{N(i)}\right)$, and similarly $KL_{Word}(N_i||F_i)$.

Figure 2 shows the largest absolute word KL-divergence differences in descending order of $|\Delta KL^i_{Word}|$ of the top words for various datasets. Positive values (of ΔKL^i_{Word}) are above x-axis and negative values (of ΔKL^i_{Word}) are below x-axis. We report the contribution of top k words to $KL(F||N)$, $KL(N||F)$, and ΔKL for k = 200 and k = 300 in Table 7. Lastly, for qualitative inspection, we also report some top words according to $|\Delta KL^i_{Word}|$ for AMT data in Table8(a) and our real-life datasets in Table 8(b) and (c) respectively. From Figure 2 and Tables 6 and 7, we can make the following crucial observations:

1. Figure 2(a) shows a somewhat "symmetric" distribution of ΔKL^i_{Word} for top words (i.e., the curves above and below $y = 0$ are equally dense) of the AMT data. This tells us that among the top words, there are two sets of words: i) set of words E which appear more in fake (than in non-fake) reviews, i.e., $\forall i \in E$, $F(i) > N(i)$ resulting in $\Delta KL^{i\in E}_{Word} > 0$ and ii) set of words, G which appear more in non-fake (than in fake) reviews, i.e., $\forall i \in G$, $N(i) > F(i)$ resulting in $\Delta KL^{i\in F}_{Word} < 0$. Moreover, the upper and lower curves being equally dense implies $|E| \approx |G|$. Additionally, the top k = 200, 300 words (see Table 7(a, b), col. 1) only contribute about 20% to ΔKL for the AMT data. This reveals that there are many more words in the AMT data which have higher probabilities in fake than non-fake reviews (i.e., the words in set E) while also many more other words which have higher probabilities in non-fake than fake reviews (i.e., the words in set G). Thus, for the AMT data, the fake and non-fake reviews consist of words with very different frequencies. This means that the Turkers didn't do a good job at "writing fake reviews" because they had little knowledge about the hotels and/or did not put their heart into writing the reviews.

   We now inspect the top words (according to $|\Delta KL_{Word}|$) for the AMT data in Table 8(a) along with their respective class probabilities[4]. Words with $\Delta KL^i_{Word} < 0$ (those having higher probabilities in non-fake than fake reviews) are marked in red. We can see that words with $\Delta KL^i_{Word} > 0$, i.e., words appearing more in fake than non-fake reviews (e.g., had, has, view, etc.) are in fact quite general words and do not show much "pretense" or "deception" as we would expect in fake reviews.
2. For our real-life spam datasets (Table 6, rows 2-5), we see that $KL(F||N)$ is much larger than $KL(N||F)$ and $\Delta KL > 1$. Figure 2 (b,…,e) also shows that among the top k = 200 words which contribute a major percentage (about 70%) to ΔKL (see Table 7 (a), row 1), most words have $\Delta KL^i_{Word} > 0$ (as the curve above y = 0 in Figure 2 is quite dense) and only few words have $\Delta KL^i_{Word} < 0$ (as the curve below $y = 0$ is very sparse). Beyond k = 200 words, we find $\Delta KL^i_{Word} \approx 0$ (except for Hotel 4-5 star whose $\Delta KL^i_{Word} \approx 0$ beyond k = 340 words). To analyze the trend, we further report contribution of top k = 300 words in Table 7 (b). We see a similar trend for k = 300 words, which shows that for our real-life data, certain top k words contribute most to ΔKL. We omit the plots for k = 300 and higher values because they too show a similar trend. Let A denote the set of those top words which contribute most to ΔKL. Further A can be partitioned into sets $A^F = \{i|\Delta KL^i_{Word} > 0\}$ (i.e., $\forall i \in A^F, F(i) > N(i)$) and $A^N = \{i|\Delta KL^i_{Word} < 0\}$ (i.e., $\forall i \in A^N, N(i) > F(i)$) where $A = A^F \cup A^N$ and $A^F \cap A^N = \phi$. Also, as the curve above $y = 0$ is dense while the curve below $y = 0$ sparse, we have $|A^F| \gg |A^N|$.

   Further, $\forall i \notin A$, we have $\Delta KL^i_{Word} \approx 0$ which implies that for $\forall i \notin A$, either one or both of the following conditions hold:

i) The word probabilities in fake and non-fake reviews are almost the same, i.e., $F(i) \approx N(i)$ resulting in $\log\left(\frac{F(i)}{N(i)}\right) \approx \log\left(\frac{N(i)}{F(i)}\right) \approx 0$ and making $KL_{Word}(F_i||N_i) \approx KL_{Word}(N_i||F_i) \approx 0$.

ii) The word probabilities in fake and non-fake are both very small, i.e., $F(i) \approx N(i) \approx 0$ resulting in very small values for $KL_{Word}(F_i||N_i) \approx 0$ and $KL_{Word}(N_i||F_i) \approx 0$, making $\Delta KL^i_{Word} \approx 0$.

These two conditions and the top words contributing a large part of ΔKL for our real-life datasets (Table 7) clearly show that in the real-life setting, most words in fake and non-fake reviews have almost the same or low frequencies (i.e., the words $i \notin A$, which have $\Delta KL^i_{Word} \approx 0$). $|A^F| \gg |A^N|$ also clearly tell us that there also exist some words which contribute most to ΔKL and which appear in fake reviews with much higher frequencies than in non-fake reviews, (i.e. the words $i \in A^F$, which have $F(i) \gg N(i)$, $\Delta KL^i_{Word} > 0$). This reveals an important insight.

- The spammers in our real-life data from Yelp made an effort (are smart enough) to ensure that their fake reviews have most words that also appear in truthful (non-fake) reviews so as to appear convincing (i.e., the words $i \notin A$ with $\Delta KL^i_{Word} \approx 0$). However, during the process of "faking", psychologically they happened to *overuse* some words with much higher frequencies in their fake reviews than in non-fake reviews (words $i \in A^F$ with $F(i) \gg N(i)$). Also, as $|A^F| \gg |A^N|$, only a small number of words are more frequent in non-fake reviews than in fake reviews. In short, the spammers seem to have "overdone faking" in pretending being truthful.

  Specifically, for our Hotel domain, some of these words in A^F with $\Delta KL_{Word} > 0$ (see Table 8(b)) are: us, price, stay, feel, nice, deal, comfort, etc. And for Restaurant domain (Table 8(c)) these include: options, went, seat, helpful, overall, serve, amount, etc. These words demonstrate marked pretense and deception. Prior works in personality and psychology research (e.g., [33] and references therein) have shown that deception/pretense usually involves more use of personal pronouns (e.g., "us") and associated actions (e.g., "went," "feel") towards specific targets ("area," "options," "price," "stay," etc.) with the objective of incorrect projection (lying or faking) which often involves more use of positive sentiments and emotion words (e.g., "nice," "deal,"

Dataset	Unique Terms	$KL(F\|\|N)$	$KL(N\|\|F)$	ΔKL	JS
Ott et al. [36]	6473	1.007	1.104	-0.097	0.274
Hotel	24780	2.228	0.392	1.836	0.118
Restaurant	80067	1.228	0.196	1.032	0.096
Hotel 4-5 ★	17921	2.061	0.928	1.133	0.125
Restaurant 4-5 ★	68364	1.606	0.564	1.042	0.105

Table 6: KL-Divergence of unigram language models.

% Contr.	Ott. et al.	Ho.	Re.	Ho. 4-5 ★	Re. 4-5 ★
ΔKL	20.1	74.9	70.1	69.8	70.3
$KL(F\|N)$	8.01	78.6	89.6	82.7	73.5
$KL(N\|F)$	5.68	15.1	12.5	12.4	17.1

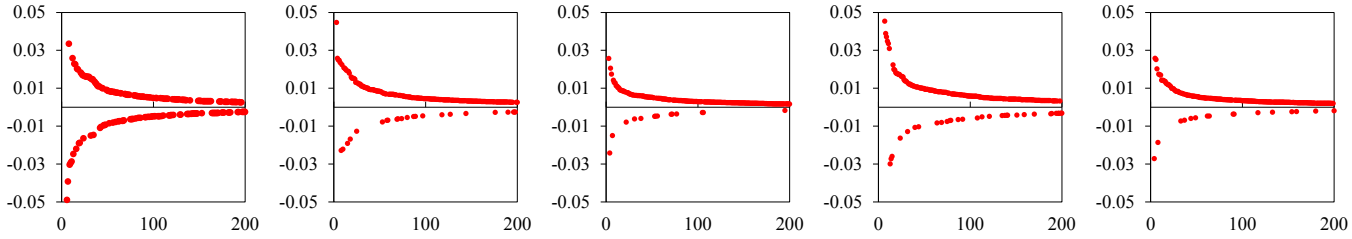
(a) k = 200

Ott. et al.	Ho.	Re.	Ho. 4-5 ★	Re. 4-5 ★
22.8	77.6	73.1	70.7	72.8
9.69	80.4	91.2	85.0	75.9
7.87	17.6	14.0	13.9	19.6

(b) k = 300

Table 7: Percentage (%) of contribution to divergence for top k words. Ho: Hotel and Re: Restaurant.

[4] As the word probabilities can be quite small, we report enough precision to facilitate accurate reproduction of results.



(a) Ott et al. [36] **(b) Hotel** **(c) Restaurant** **(d) Hotel 4-5 ★** **(e) Restaurant 4-5 ★**

Figure 2: Word-wise difference of KL-Div (ΔKL_{Word}) across top 200 words (using $|\Delta KL_{Word}|$) for different datasets.

"comfort," "helpful," etc.).

Now let us go back to the AMT data. From Table 6, 7 and 8(a) and the similar size of $|A^F|$ and $|A^N|$, we can conclude:

- AMT fake reviews use quite different words than the genuine reviews. This means the Turkers didn't do a good job at faking, which is understandable as they have little gain[5] in doing so.

We now also throw some light on the contribution of top k words to KL, ΔKL in Table 7.

1. For our real-life data, top k (= 200, 300) words contribute about 75-80% to ΔKL showing that spammers do use specific deception words. For AMT data, we find very little contribution (≈ 20%) of those top k words towards ΔKL, i.e., there are many other words which contribute to ΔKL resulting in very different set of words in fake and non-fake reviews (which also explains the reason for higher JS-Div. for AMT data in Table 6).
2. Further from Table 7 (rows 2, 3), for real-life spam data, we find that top k words contribute almost 80-90% to $KL(F||N)$. This shows that spammers use specific pretense/deception words more frequently to inflict opinion spam. It is precisely these words which are responsible for the deviation of the word distribution of spammers from non-spammers. However for $KL(N||F)$, we see very low contribution of those top words in our real-life data. This reveals that genuine reviews do not tend to purposely use any specific words. Psychologically, this is true in reality because to write a genuine experience, people do not need any typical words but just state the true experience.

We also conducted experiments on bigram distributions and different k which too yielded similar trends for KL and ΔKL like unigram models but resulted in smaller KL and higher JS values because with bigrams the term space gets sparser with net probability mass being distributed among many more terms.

To summarize, let us discuss again why the real-life spam dataset is much harder to classify than the AMT data. Clearly, the symmetric Jensen–Shannon (JS) divergence results (JS col. in Table 6) show that the JS values for our real-life datasets are much lower (almost half) than that for the AMT data (JS divergence is bounded by 1, $0 \leq JS \leq 1$, when using $\log_2$). This implies that fake and non-fake reviews in the AMT data are easier to separate/classify than in our real-life data. This is also shown from our analysis results above as in the AMT data, fake and non-fake reviews use very different word distributions (resulting in a higher JS-Div.), while for our data, the spammers did a good job (they knew their domains well) by using those words which appear in non-fake reviews in their fake reviews almost equally frequently. They only overuse a small number of words in fake reviews due to (probably) trying too hard to make them sound real. However, due to the small number of such words, they may not appear in every fake review, which again explains that fake and non-fake reviews are much harder to separate or to classify for our real-life datasets.

Lastly, we note that the trends of 4-5 star reviews from popular hotels and restaurants are the same as that of the entire data, which indicates that the large accuracy difference between AMT and Yelp data are mainly due to the fake review crafting process rather than that [36] used only positive reviews from popular hotels.

Word (w)	ΔKL_{Word}	P(w|*F*) (in E-4)	P(w|*N*) (in E-6)
were	-0.147	1.165	19822.7
we	-0.144	1.164	19413.0
had	0.080	163.65	614.65
night	-0.048	0.5824	7017.36
out	-0.0392	0.5824	5839.26
has	0.0334	50.087	51.221
view	0.0229	36.691	51.221
enjoyed	0.0225	44.845	153.664
back	-0.019	0.5824	3226.96
felt	0.0168	28.352	51.222

(a) Ott et al. [36]

Word (w)	ΔKL_{Word}	P(w|*F*) (in E-4)	P(w|*N*) (in E-6)
us	0.0446	74.81	128.04
area	0.0257	28.73	5.820
price	0.0249	32.80	17.46
stay	0.0246	27.64	5.820
said	-0.0228	0.271	3276.8
feel	0.0224	24.48	5.820
when	-0.0221	55.84	12857.1
nice	0.0204	23.58	5.820
deal	0.0199	23.04	5.820
comfort	0.0188	21.95	5.820

(b) Hotel

Word (w)	ΔKL_{Word}	P(w|*F*) (in E-4)	P(w|*N*) (in E-6)
places	0.0257	25.021	2.059
options	0.0130	12.077	0.686
evening	0.0102	12.893	5.4914
went	0.0092	8.867	0.6864
seat	0.0089	8.714	0.6852
helpful	0.0088	8.561	0.6847
overall	0.0085	8.3106	0.6864
serve	0.0081	10.345	4.8049
itself	-0.0079	.10192	1151.82
amount	0.0076	7.542	0.6864

(c) Restaurant

Table 8: Top words according to $|\Delta KL_{Word}|$ with their respective *F*ake/*N*on-fake class probabilities P(w|*F*) (in E-4, i.e., 10^{-4}), P(w|*N*) (in E-6, i.e., 10^{-6}) for different datasets.

4. CAN AMT FAKE REVIEWS HELP IN DETECTING REAL FAKE REVIEWS?

An interesting question is: Can we use the AMT fake reviews to detect real-life fake reviews? This is important because not every website has filtered reviews that can be used in training. When there are no reliable ground truth fake and non-fake reviews for model building, can we employ crowdsourcing (e.g., Amazon Mechanical Turk) to generate fake reviews to be used in training? To answer this question, we conduct the following experiments:

Setting 1: Train using the original 400 fake reviews from AMT and 400 non-fake reviews from Tripadvisor and test on 4-5 star Yelp reviews from the same 20 hotels as those used in [36].

Setting 2: Train using the 400 AMT fake reviews in [36] and randomly sampled 400 4-5 star unfiltered (non-fake) Yelp reviews from the same 20 hotels, and test on fake and non-fake 4-5 star Yelp reviews from the 20 hotels.

Setting 3: Train exactly as in Setting 2, but test on fake and non-fake 4-5 star reviews from all our Yelp hotel domain data *except* those 20 Hotels. Here we want to see whether the classifier built using the reviews from the 20 hotels can be applied to other hotels. After all, it is quite hard and expensive to use AMT to generate fake reviews for every individual hotel before a classifier can be applied to the hotel.

For training we use balanced data for better classification results (see § 2.2.1 and 2.2.3) and for testing we again have two settings: balanced data (50:50) and natural distribution (N.D.) as before. The results of the three settings are given in Table 9. We make the following observations:

1. For the real-life balanced test data, both the accuracy and F1 scores are much lower than those from training and testing

[5] The Turkers are paid only US$1 per review and they do not have genuine interest to write fake reviews (because they are hired for research). However, the real fake reviewers on Yelp both know the domain/business well and also have genuine interests in writing fake reviews for that business in order to promote/demote.

using only the AMT data (Table 5(a)). Accuracies in the 50:50 setting in Table 9 indicate near chance performance.

2. For the real-life natural distribution test data, the F1 score and recall (for fake reviews) is also dramatically lower than those from training and testing using the real-life Yelp data (Table 5(b)). The accuracies are higher because of the skewed data, but very low recall implying poor performance on detection.

In summary, we conclude that model trained on AMT generated fake reviews are weak in detecting fake reviews in real-life and detection accuracies are near chance. This indicates that the fake reviews from AMT are not representative of real fake reviews on Yelp. Note that we only experimented with the hotel domain because the AMT fake reviews in [36] are only for hotels.

5. BEHAVIORAL ANALYSIS

The previous sections performed hardness analysis of fake review classification using linguistic features (i.e., *n*-grams). A natural question is: can we improve the classification on the real-life dataset under its natural class distribution (N.D.)? The answer is yes. This section proposes some highly discriminating spamming behavioral features to demarcate spammers and non-spammers. We will see that these behaviors can help improve classification dramatically. More interestingly, these features alone are actually able to do much better than *n*-gram features.

5.1 Behavioral Features and Analysis

For the behavioral study, we crawled profiles of all reviewers in our hotel and restaurant domains. We present the behavioral features about reviewers below and at the same time analyze their effectiveness. Since we analyze each reviewer, to facilitate the analysis, we separate reviewers in our data (Table 1) into two groups: (1) spammers: those who wrote fake (filtered) reviews in our data and (2) non-spammers: those who did not write fake (filtered) reviews in our data. Note that we do not claim that these non-spammers have not spammed on other businesses, or the spammers do not have any truthful reviews on other businesses. We only refer the reviewers to the above terminology based on our data in Table 1 only. Nevertheless, it is important to note that our data yielded 8033 spammers and 32684 non-spammers showing that about 20% of reviewers are spammers in our data.

1. Activity Window (AW): Fake reviewers are likely to review in short bursts and are usually not longtime active members [25, 51]. Genuine reviewers on the other hand are people who mostly write true experiences with reviews posted from time to time. It is interesting to measure the activity freshness of accounts (reviewer-ids). For reviewers in our data, we compute the feature, *activity window* as the difference of timestamps of the last and first reviews for that reviewer. We plot the cumulative distribution function (CDF) curve of activity window in months for spammers and non-spammers in Figure 3(a). A majority (80%) of spammers are bounded by 2 months of activity whereas only 20% of non-spammers are active for less than 10 months (i.e., 80% of non-spammers remain active for at least 10 months).

2. Maximum Number of Reviews (MNR): In our data, we found that 35.1% of spammers posted all their reviews in a single day. Naturally, the maximum number of reviews in a day is a good feature. The CDF of this MNR feature in Figure 3(b) shows that only 25% of spammers are bounded by 5 reviews per day, i.e., 75% of spammers wrote 6 or more reviews per day. Non-spammers have a very moderate reviewing rate (50% write 1 review per day and 90% are bounded by 3 reviews per day).

3. Review Count (RC): This is the number of reviews that a reviewer has. Activity window showed that spammers are not longtime members which reflects that they probably also have fewer reviews as they are not really interested in reviewing, but just interested in promoting certain businesses. We show the CDF of number of reviews for spammers and non-spammers in Figure 3(c). We note that 80% of spammers are bounded by 11 reviews. However, for non-spammers, only 20% are bounded by 20 reviews, 50% are bounded by 40 reviews, and the rest 50% have more than 40 reviews. This shows a clear separation of spammers from non-spammers based on their reviewing activities.

n-gram	C.D.	P	R	F1	A
Unigram	50:50	57.5	31.0	40.3	52.8
	N.D.	13.5	31.0	18.8	73.1
Bigram	50:50	57.3	31.8	40.9	53.1
	N.D.	13.1	31.8	18.6	72.8

(a) Setting 1

P	R	F1	A
62.1	35.1	44.9	54.5
14.2	35.1	20.2	77.5
62.8	35.3	45.2	54.9
14.0	35.3	20.0	76.9

(b) Setting 2

P	R	F1	A
67.3	32.3	43.7	52.7
19.7	32.3	24.5	78.7
67.6	32.2	43.6	53.2
19.2	32.2	24.0	78.0

(c) Setting 3

Table 9: SVM 5-fold CV classification results using AMT generated 400 fake hotel reviews as the positive class in training.

4. Percentage of Positive Reviews (PR): Opinion spamming can be used for both promotion and demotion of target businesses. This feature is the percentage of positive (4 or 5 star) reviews. We plot the CDF of percentage of positive reviews among all reviews for spammers and non-spammers in Figure 3(d). We see that about 15% of the spammers have less than 80% of their reviews as positive, i.e., a majority (85%) of spammers has more than 80% of their reviews being positive. Non-spammers on the other hand show a rather evenly distributed trend where we find a varied range of reviewers who have different percentage of 4-5 star reviews. This is reasonable because in real-life, people (genuine reviewers) may have different levels of rating nature.

5. Review Length (RL): As opinion spamming involves writing fake experiences, there is probably not much to write or at least a (paid) spammer probably does not want to invest too much time in writing. We show the CDF of the average number of words per review for all reviewers in Figure 3(e). We see that a majority (≈ 80%) of spammers are bounded by 135 words in average review length which is quite short as compared to non-spammers where we find only 8% are bounded by 200 words while a majority (92%) have higher average review word length (> 200).

6. Reviewer deviation (RD): This feature analyzes the amount that spammers deviate from the general rating consensus. To measure reviewer deviation, we first compute the absolute rating deviation of a review from other reviews on the same business. Then, we compute the average deviation of a reviewer by taking the mean of all rating deviations over all his reviews. On a 5-star scale, the deviation can range from 0 to 4. We plot the CDF of spammers and non-spammers for reviewer deviation in Figure 3(f). We find that most non-spammers (≈ 70%) are bounded by an absolute deviation of 0.6 on a 5-star scale which shows that non-spammers have rating consensus with other genuine reviewers for a business. However, only 20% of spammers have deviation less than 2.5 and most spammers deviate a great deal from the rest.

7. Maximum content similarity (MCS): Crafting a new fake review every time is time consuming. To examine whether some posted reviews are similar to previous reviews, we compute the maximum content similarity (using cosine similarity) between any two reviews of a reviewer. Figure 3(g) shows its CDF plot. About 70% of non-spammers have very little similarity (bounded by 0.16 cosine similarity) across their reviews showing that most non-spammers write reviews with new content. On the other hand, we see about 30% of spammers are bounded by a cosine similarity of 0.3 and the rest 70% have a lot of similarity across their reviews. This shows that content similarity is another metric where opinion spamming is exhibited quantitatively.

8. Tip Count (TC): Yelp prohibits review posts via mobile devices. However, it facilitates "tip" submissions via mobile devices. Tips are short (140 characters) descriptions and insights about a business which facilitate reviewers to catalog their

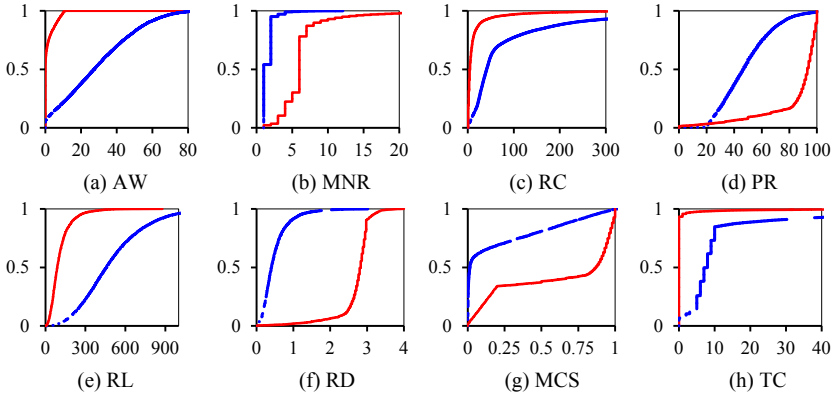


Figure 3: CDF of Behavioral Features. Cumulative percentage of spammers (in red/solid) and non-spammers (in blue/dotted) vs. behavioral feature value.

immediate experiences that can be expanded later [7]. We investigate the tip counts for spammers and non-spammers. The CDF shows a clear separation (Figure 3(h)). About 90% of spammers posted 0 tips while only 15% of non-spammers are bounded by 5 tips (i.e., 85% of non-spammers posted more than 5 tips). This shows that spammers mostly do not post tips.

The above analysis shows that the proposed features are quite discriminating. There are also various other metadata that can be extracted from Yelp which may be useful in identifying fake reviews. These include friendship and fan relations, compliments and usefulness votes, percentage of previous reviews filtered, etc. However, using these features for classification is not fair because they are somewhat directly or indirectly affected by Yelp's filtering, e.g., if a review is filtered, its chance of getting usefulness votes, compliments, friend and fan requests reduce automatically. Our proposed features above are not affected or at most minimally affected by Yelp's filtering. Furthermore, in individual feature experiments in § 6, we will see that dropping any feature will not make a major difference in classification performances.

5.2 Statistical Validation

The previous sub-section presented the behavioral features and reported the relative discriminative strengths of different behaviors across spammers and non-spammers. In § 6, we will use these reviewer behavioral features for classification of reviews. Before proceeding, it is useful to study the correlation of fake reviews and abnormal reviewing behaviors in § 5.1. This study also indirectly verifies the quality of the Yelp's filtered review labels.

We first normalize all behavioral features by the maximum value in our data so that they are continuous features in [0, 1]. As MNR, RC, and RL features are never 0 (they are at least 1), we subtract 1 from these features prior to normalization to ensure they lie within [0, 1]. Further to ensure that values close to 1 indicate spamming, we use the flipped versions for four behaviors: AW = 1 – AW, RC = 1 – RC, RL = 1 – RL, and TC = 1 – TC as lower values in these features indicate spamming.

Formally, for a given reviewer behavior f, its effectiveness ($Eff(\cdot)$) across fake and non-fake reviews can be defined as follows: $Eff(f) \equiv P(f > \beta|Fake) - P(f > \beta|Nonfake)$ where $f > \beta$ is the event that the corresponding behavior exhibits spamming. On a scale of [0, 1] where values close to 1 (respectively 0) indicate spamming (non-spamming) choosing a threshold β is somewhat subjective. While $\beta = 0.5$ is reasonable (as it is the expected value of variables uniformly distributed in [0, 1]), $\beta = 0$ is very strict, and $\beta = 0.25$ is rather midway. We experiment with all three threshold values for β. Let the *null hypothesis* be: Reviewers of both fake and non-fake reviews are equally likely to exhibit f (spamming or attaining values > β), and the *alternate hypothesis*: reviewers of fake reviews are more likely to exhibit f than reviewers of non-fake reviews and are correlated with f. Thus, demonstrating that reviewer behavior f is correlated with fake reviews is reduced to showing that $Eff(f) > 0$. A Fisher's exact test[6] rejects the *null hypothesis* with $p<0.01$ across different threshold values β for each of the behaviors. This shows that fake (filtered) reviews are indeed correlated with abnormal behaviors of their corresponding reviewers. Furthermore, since the features are all *anomalous* and not directly linked with filtering, and Fisher's exact test verifies strong correlation of those reviewer behaviors with reviews which were "filtered", it also indirectly gives us a strong confidence that the vast majority of the class labels (fake: filtered, non-fake: unfiltered) in the Yelp dataset are trustworthy.

6. USING BEHAVIORAL FEATURES IN CLASSIFICATION

We now report fake review classification using reviewer behavioral features. For each review we add the behavioral feature of its reviewer. We report the 5-fold SVM classification results across various feature settings in Table 10. Again to ensure effective learning, we use balanced data (50:50) for training (see § 2.2.1 and 2.2.2), but both balanced data and natural distribution data for testing. We note the following:

1. Using only behavioral features (BF) boosts precision by about 20% and recall by around 7% in both domains resulting in around 14% (in 50:50) and 24% (in N.D.) improvement in F1. The much higher F1 for the N.D. setting is very noteworthy.
2. Behavioral features (BF) alone perform much better than only *n*-grams, which shows that behavioral features are stronger than linguistic *n*-grams for opinion spam (fake review) detection.
3. *n*-grams + BF improves F1 slightly by about 3% beyond using only BF. Compared with the results in rows 1, 2 and 3, we can see that the gain is mainly attributed to BF.
4. Although behavioral features render a substantial boost in F1, the problem of detecting opinion spam still remains to be very hard in the natural distribution of the realistic situation as the model only obtains about 60% in F1 score.

Note that we cannot use behavioral features for the AMT data [36] as its fake reviews are generated by Turkers with no behavior information, which is a drawback of the AMT data.

Effect of individual features: We now perform some additional experiments to investigate the contribution of each behavioral feature. Table 10 shows that Bigrams+BF gives the highest accuracy and F1-score for the 50:50 setting and slightly lower accuracy and F1 for the N.D. setting (than Unigrams+BF). Hence, we drop a behavioral feature at a time from the full feature set Bigrams+BF. We report results in Table 11. Note that feature selection metrics, e.g., Information Gain (IG) can also be used to assess the relative strength of each behavioral feature. However, IG of a feature only reports the net reduction in entropy when that feature is used to partition the data. Although reduction in entropy using a feature (i.e., gain obtained using that feature) is correlated with the discriminating strength of that feature, it does not give any indication on the actual performance loss when that feature is dropped. Here, we want to understand the effect of each feature.

[6] χ^2 test could also have been be used. However, for skewed data (like ours) the exact and asymptotic *p*-values can be quite different leading to opposite conclusions regarding the hypothesis because the sampling distribution of the test statistic only approximates the theoretical χ^2distribution [28].

Apart from dropping individual features, we further drop AW, RC, and TC together (see last row, Table 11). The rationale here is that when a reviewer sees most of his review filtered by Yelp, he is probably going to abandon that account which eventually will result in low values for AW, RC, and TC as the account is no longer in use. However, MNR, PR, RL, RD, and MCS "recorded" past behaviors which cannot be undone and has to do with the very reviewing nature per-se than the account's reviews being filtered.

Table 11 shows that dropping individual behavioral features results in a graceful degradation in performance across both 50:50 and N.D. settings for hotel and restaurant domains. Dropping features AW, PR, RL, MCS, and TC result in about 4-6% reduction in accuracy and F1-scores in 50:50 and N.D. setting showing that those features are more useful for classification. Dropping other features also results in 2-3% performance reduction. Dropping AW, RC, and TC features results in about 6-8% drop in F1 than the BF setting (Table 10, row 3). This shows that all the behavioral features are useful for fake review detection. Furthermore, even with reduced feature set (i.e., dropping one feature at a time, or dropping AW, RC, and TC features together), the model significantly outperforms textual *n*-grams. This is quite promising. We believe that our framework should be generic and applicable for fake review detection in other online review websites (as all features except "tip count" can be computed using posting date and star rating, which are almost always available). Although the exact *results* obtained on Yelp may not directly apply to other sites, as reviewer activities can be different across different sites [50]; the behaviors are general and can be adapted for other sites.

7. QUALITY OF YELP FILTERING

After all the experiments, we now come back to discuss the quality of Yelp filtering again because all our analyses hinge on the quality of Yelp data. We want to make the following claims and discuss its impact on our analyses and classifications:

1. **Yelp's filtering precision is good**. We have three evidences to support this claim: (1) Classification under the balance distribution showed an accuracy of 67.8%, which is significantly higher than random guessing of 50%. This indicates that linguistically there is significant difference between filtered and unfiltered reviews which implies different psychological states of the minds of the two groups of reviewers when they write reviews. (2) Using abnormal behaviors render even higher accuracy. It is abnormal for a genuine reviewer to exhibit those behaviors. (3) Yelp's filtering has been there for years. Although there are some complaints about filtering genuine reviews, considering the huge number of filtered reviews in Yelp some false positives are acceptable. If Yelp's filtering is really random, we believe that Yelp would not have it used for the past 6-7 years (they started to work on filtering shortly after their launch in 2004 [46]). Although these are not hard evidences, they do render confidence that Yelp is doing a reasonable job and its filtering is sufficiently reliable.

2. **It is hard to know the recall**. We have no evidence about how good the recall is. However, it should be safe to say that the proportion of fake reviews in the unfiltered set is small. Then, they will not affect the probability distributions much. Our analysis in § 3, 5.1, 5.2 is still valid. If Yelp catches more fake reviews (higher recall) for use in training, the classification results will improve.

3. How does Yelp filter? Although an interesting question, it is hard to know the exact clues that Yelp uses. However, from our results in § 6, we can speculate that Yelp probably uses some behaviors and also many internal metrics (e.g., IP addresses, session/user logs, etc.) and social network of user interactions [50] which are not publicly available for model building. This is probably the reason why our methods, although effective, still have room for improvement.

Feature Setting	C.D.	P	R	F1	A
Unigrams	50:50	62.9	76.6	68.9	65.6
	N.D.	20.3	76.6	31.9	57.4
Bigrams	50:50	61.1	79.9	69.2	64.4
	N.D.	19.8	79.9	31.7	54.6
Behavior Features (BF)	50:50	82.4	85.2	**83.7**	**83.8**
	N.D.	41.4	84.6	**55.6**	82.4
Unigrams + BF	50:50	83.8	81.4	82.5	84.0
	N.D.	47.5	80.6	**59.8**	85.5
Bigrams + BF	50:50	86.9	82.8	**84.8**	**85.1**
	N.D.	46.5	82.5	59.4	84.9

(a): Hotel

P	R	F1	A
64.3	76.3	69.7	66.9
20.4	76.3	32.2	60.1
64.5	79.3	71.1	67.8
20.2	79.3	32.1	58.6
82.8	88.5	**85.6**	**83.3**
48.2	87.9	**62.3**	78.6
83.9	87.6	85.7	84.7
49.9	87.1	**63.4**	82.6
84.5	87.8	**86.1**	**86.5**
48.9	87.3	62.7	82.3

(b): Restaurant

Table 10: SVM 5-fold CV classification results across behavioral (BF) and *n*-gram features, P: Precision, R: Recall, F1: F1-Score on the fake class, A: Accuracy. Training uses balanced data (50:50). Testing uses two class distributions (C.D): 50:50 (balanced) and Natural Distribution (N.D.). Improvements using behavioral features over unigrams and bigrams are statistically significant with p<0.005 based on paired *t*-test.

Dropped Feature	C.D.	P	R	F1	A
AW	50:50	82.0	79.1	80.5	78.9
	N.D.	42.3	79.1	55.1	80.1
MNR	50:50	84.9	80.6	82.7	83.3
	N.D.	43.8	80.6	56.8	82.5
RC	50:50	84.5	80.0	82.2	82.8
	N.D.	43.2	80.0	56.1	82.0
PR	50:50	82.9	78.2	80.5	80.1
	N.D.	43.0	78.2	55.5	81.3
RL	50:50	82.7	78.0	80.3	79.7
	N.D.	43.2	78.0	55.6	80.0
RD	50:50	85.2	81.6	83.4	84.0
	N.D.	44.2	81.6	57.3	83.2
MCS	50:50	83.9	80.1	81.9	82.9
	N.D.	44.2	80.1	56.9	81.0
TC	50:50	82.7	80.0	81.3	80.2
	N.D.	43.1	80.0	56.0	80.9
AW, RC, TC	50:50	76.9	78.9	77.9	77.4
	N.D.	35.2	78.9	48.7	76.0

(a): Hotel

P	R	F1	A
79.9	82.2	81.0	81.8
44.0	82.2	57.3	77.9
82.8	86.0	84.4	84.4
47.0	86.0	60.8	80.0
82.4	86.3	84.3	84.7
47.6	86.3	61.4	81.9
81.3	83.4	82.3	82.5
45.2	83.4	58.6	77.9
81.8	82.9	82.3	81.8
45.6	82.9	58.8	78.6
83.4	86.7	85.0	85.7
48.1	86.7	61.9	81.9
82.8	85.0	83.9	84.3
46.2	85.0	59.9	80.1
80.7	83.9	82.3	83.4
45.2	83.9	58.7	79.8
74.4	78.6	76.4	74.1
46.1	78.6	58.1	73.9

(b): Restaurant

Table 11: SVM 5-fold CV classification results by dropping behavioral features from the full feature set Bigram+BF (Table 10, last row). Differences in classification metrics for each dropped feature are statistically significant with p<0.01 based on paired *t*-test.

8. CONCLUSIONS

This paper performed an in-depth investigation of supervised learning for fake review detection using Amazon Mechanical Turk (AMT) generated fake reviews and real-life fake reviews. The work in [36] showed that using AMT fake reviews and reviews (assumed non-fake) from Tripadvisor achieved the classification accuracy of 89.6% with bigram features and balanced data. This paper first performed a comparison using real-life filtered (fake) and unfiltered (non-fake) reviews in Yelp. The results showed that the real-life data is much harder to classify, with an accuracy of only 67.8%. This prompted us to propose a novel and principled method to uncover the precise difference between the two types of fake reviews using KL-divergence and its asymmetric property. Our analysis showed that the Turkers didn't do a good job at faking. Furthermore, models trained using AMT fake reviews are weak in detecting real fake reviews in Yelp, which indicates that the AMT fake reviews are probably not representative of the real-life fake reviews. To improve classification on Yelp's real-life data, a set of behavioral features were proposed which resulted in a major accuracy improvement. Also interesting is the fact that behavioral features alone perform better than *n*-gram features.

9. REFERENCES

[1] Castillo,C.,Donato,D.,Gionis,A.,Murdock,V.,Silvestri,F. Know your neighbors: web spam detection using the web topology. SIGIR 2007: 423-430

[2] Chawla,N.,Japkowicz,N. and Kolcz,A. Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets, 6(1):1–6, 2004

[3] Cheng,Z.,Gao,B.,Sun,C.,Jiang,Y.,Liu,T. Let web spammers expose themselves. WSDM 2011: 525-534

[4] Chirita, P.A., Diederich, J., and Nejdl, W. MailRank: using ranking for spam detection. CIKM. 2005.

[5] Chung,Y.,Toyoda,M.,Kitsuregawa,M. Detecting Hijacked Sites by Web Spammer Using Link-Based Algorithms. IEICE Transactions 93-D(6): 1414-1421 (2010)

[6] Drummond,C. and Holte,R. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling. In Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, 2003.

[7] Eric, Ask Yelp: Why can't I write reviews from my mobile? http://officialblog.yelp.com/2009/12/ask-yelp-why-cant-i-write-reviews-from-my-mobile.html. Yelp Official Blog. December 2009.

[8] Feng, S., Banerjee, R., and Choi, Y. Syntactic Stylometry for Deception Detection. ACL (short paper), 2012.

[9] Feng, S., Xing, L., Gogar, A., and Choi, Y. Distributional Footprints of Deceptive Product Reviews. In ICWSM. 2012.

[10] Holloway, D. Just Another Reason Why We Have a Review Filter. http://officialblog.yelp.com/2011/10/just-another-reason-why-we-have-a-review-filter.html. Yelp Official Blog. October 2011.

[11] Jindal, N. and Liu, B. Opinion spam and analysis. WSDM. 2008.

[12] Jindal, N., Liu, B. and Lim, E. P. Finding Unusual Review Patterns Using Unexpected Rules. CIKM. 2010.

[13] Jin,X.,Lin,C.X.,Luo,J.,Han,J. SocialSpamGuard: A Data Mining-Based Spam Detection System for Social Media Networks. PVLDB 4(12): 1458-1461 (2011)

[14] Joachims, T. Making large-scale support vector machine learning practical. Advances in Kernel Methods. MIT Press. 1999.

[15] Joachims, T. (1998) Text categorization with support vector machines: Learning with many relevant features.. ECML, 1998.

[16] Kang, H., Wang, K., Soukal, D., Behr, F., Zheng, Z.: Large-scale Bot Detection for Search Engines. In: Proc. of WWW, 2010.

[17] Kim, S.M., Pantel, P., Chklovski, T. and Pennacchiotti, M. Automatically assessing review helpfulness. EMNLP. 2006.

[18] Kolari, P., Java, A., Finin, T., Oates, T., Joshi, A. Detecting Spam Blogs: A Machine Learning Approach. AAAI. 2006.

[19] Kost, A. Woman Paid to Post Five-Star Google Feedback. http://www.thedenverchannel.com/news/woman-paid-to-post-five-star-google-feedback. ABC 7 News. May 2012.

[20] Koutrika, G., Effendi, F. A., Gyöngyi, Z., Heymann, P., and H. Garcia-Molina. Combating spam in tagging systems. AIRWeb. 2007.

[21] Kraft,L.G. A device for quantizing, grouping, and coding amplitude-modulated pulses. Diss. Massachusetts Institute of Technology, 1949.

[22] Lauw,H.W.,Lim,E.P.,Wang,K. Bias and Controversy: Beyond the Statistical Deviation. SIGKDD 2006

[23] Lauw,H.W.,Lim,E.P.,Wang,K. Summarizing Review Scores of Unequal Reviewers. SIAM International Conference on Data Mining 2007.

[24] Li, F., Huang, M., Yang, Y. and Zhu, X. Learning to identify review Spam. IJCAI. 2011.

[25] Lim, E. Nguyen, V. A., Jindal, N., Liu, B., and Lauw, H. Detecting Product Review Spammers Using Rating Behavior. CIKM. 2010.

[26] Liu, J., Cao, Y., Lin, C., Huang, Zhou, M. Low-quality product review detection in opinion summarization. EMNLP, 2007.

[27] Luther, Yelp's Review Filter Explained. http://officialblog.yelp.com/2010/03/yelp-review-filter-explained.html. Yelp Official Blog. March 2010.

[28] Mehta, C. R, Patel, N. R, Tsiatis, A. A. Exact significance testing to establish treatment equivalence with ordered categorical data. Biometrics. 1984

[29] Mishne,G.,Carmel,D.,Lempel,R. Blocking Blog Spam with Language Model Disagreement. AIRWeb pages 1-6, 2005.

[30] Moghaddam,S.,Jamali,M.,Ester,M. ETF: extended tensor factorization model for personalizing prediction of review helpfulness. WSDM 2012: 163-172

[31] Mukherjee,A. and Liu,B. Improving gender classification of blog authors. EMNLP, 2010.

[32] Mukherjee,A.,Liu,B.,Glance,N. Spotting fake reviewer groups in consumer reviews. WWW. 2012.

[33] Newman, M.L.,Pennebaker,J.W.,Berry,D.S.,Richards,J.M. Lying words: predicting deception from linguistic styles, Personality and Social Psychology Bulletin 29, 665–675. 2003.

[34] Nisen, M. Fake Reviews Are Becoming An Even Bigger Problem For Businesses. http://www.businessinsider.com/fake-reviews-are-becoming-a-huge-problem-for-businesses-2012-9, Sep. 19, 2012.

[35] Ntoulas,A.,Najork,M.,Manasse,M.,Fetterly,D. Detecting spam web pages through content analysis. WWW 2006: 83-92

[36] Ott, M., Choi, Y., Cardie, C. Hancock, J. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. ACL. 2011.

[37] Pennebaker,J.W.,Chung,C.K.,Ireland,M.,Gonzales,A. and Booth,R.J. 2007. The development and psychometric properties of LIWC2007. Austin, TX, LIWC.net.

[38] Popken, B. 30 Ways You Can Spot Fake Online Reviews. http://consumerist.com/2010/04/14/how-you-spot-fake-online-reviews/. The Consumerist. April 2010.

[39] Saito,H.,Toyoda,M.,Kitsuregawa,M.,Aihara K. A Large-Scale Study of Link Spam Detection by Graph Algorithms (S). AIRWeb 2007

[40] Silva, J., Dias, F., Guillore, S., Lopes, G. Using LocalMaxs Algortihm for the Extraction of Contiguous and Noncontiguous Multiword Lexical Units. Springer Lecture Notes in AI 1695, 1999.

[41] Song,Y.,Kolcz,A.,Giles,L.C. Better Naive Bayes classification for high-precision spam detection. Softw., Pract. Exper. 39(11): 1003-1024 (2009)

[42] Spirin,N. and Han,J. "Survey on web spam detection: principles and algorithms." ACM SIGKDD Explorations Newsletter 13.2 (2012): 50-64.

[43] Streitfeld, D. Buy Reviews on Yelp, Get Black Mark. http://www.nytimes.com/2012/10/18/technology/yelp-tries-to-halt-deceptive-reviews.html. New York Times. October 2012.

[44] Streitfeld, D. Fake Reviews, Real Problem. http://query.nytimes.com/gst/fullpage.html?res=9903E6DA1E3CF933A2575AC0A9649D8B63. New York Times. September 2012.

[45] Streitfeld, D. In a Race to Out-Rave, 5-Star Web Reviews Go for $5. http://www.nytimes.com/2011/08/20/technology/finding-fake-reviews-online.html. New York Times. August 2011.

[46] Stoppelman, J. Why Yelp has a Review Filter. http://officialblog.yelp.com/2009/10/why-yelp-has-a-review-filter.html. Yelp Official Blog. October 2009.

[47] Sun, Y., Kamel, M., Wang, Y. 2006. Boosting for learning multiple classes with imbalanced class distribution. ICDM, 2006.

[48] Taylor, J. Are You Buying Reviews For Google Places? http://www.localgoldmine.com/blog/reputation-management/are-you-buying-reviews-for-google-places/, January 27, 2012.

[49] Wang, G., Xie, S., Liu, B., and Yu, P. S. Review Graph based Online Store Review Spammer Detection. ICDM. 2011.

[50] Wang,Z. Anonymity, social image, and the competition for volunteers: a case study of the online market for reviews. The BE Journal of Economic Analysis & Policy. 2010.

[51] Weise. K. A Lie Detector Test for Online Reviewers. http://www.businessweek.com/magazine/a-lie-detector-test-for-online-reviewers-09292011.html. BusinessWeek. September 2011.

[52] Xie,S.,Wang,G.,Lin,S.,Yu,P.S. Review spam detection via temporal pattern discovery. KDD 2012: 823-831

[53] Yang,H.,King,I.,Lyu,M.R. DiffusionRank: a possible penicillin for web spamming. SIGIR 2007: 431-438

[54] Zhang, Z. and Varadarajan, B. Utility scoring of product reviews. CIKM. 2006.

[55] Zhou, B. Pei,J. Link spam target detection using page farms. TKDD 3(3).2009.