Detecting Deceptive Opinion Spam

Kent Blake
Stanford University
CS 224U
kmblake@stanford.edu

Allison Dods
Stanford University
CS 224U
adods@stanford.edu

Alec Glassford
Stanford University
CS 224U
gla@stanford.edu

Abstract

Consumers who base purchase decisions on online reviews must be able to trust those reviews. However, humans are notably bad at distinguishing between genuine reviews and deceptive ones, which might be posted by companies to inflate their own reputations or lower their competitors. This study replicates recent work in detecting deceptive online reviews using machine learning techniques. We present results comparable to those in the recent literature, using a variety of models and features. Most significantly, we contribute a brief exploration of the use of shallow neural networks in deception detection. Our results suggest that neural networks are a promising direction for future work. Finally, we find that, despite their general success, the models used in the current study struggle with accurately classifying reviews with verifiable claims and those that reference other reviewers.

1 Introduction

One problem for websites that collect or display online product reviews, such as Yelp, Amazon, or TripAdvisor, is the proliferation of deceptive review spam. In this paper, we investigate methods for distinguishing between truthful and deceptive reviews. Here, a "truthful" review is defined as one written genuinely and which describes the reviewer's authentic experience using the product. "Deceptive" reviews, on the other hand, are those that do not accurately convey a person's actual experience using a product.

Reviewers might post deceptive reviews for a number of reasons. In some cases, deceptive reviews are written by people who have never used the product in question; in others, they are written by people who are familiar with the product but who are fabricating their opinions of it. For instance, a company might fabricate a positive review of its own product to boost its reputation; fabricate a negative review of a competitor's product to lower its rating; or bribe product users to post deceptively on its behalf [Li et al. (2014)].

Distinguishing truthful from deceptive reviews benefits both consumers and review-aggregating websites. Consumers who wish to purchase a product, eat at a restaurant, stay at a hotel, or otherwise use a service based on online reviews must be able to trust those reviews. In a similar vein, websites that are able to filter deceptive reviews may be seen as more reliable. However, humans are quite bad at detecting deceptive reviews; Ott and colleagues found that of three human judges, the highest accuracy score achieved on a deception detection test was 65% [2013]. Therefore, researchers have turned to machine learning-based techniques to solve the problem.

The current study replicates the performance of several recent studies using a variety of models and features. It contributes to the literature an attempt to identify deceptive reviews using neural networks, specifically a shallow multi-layer perceptron model. We find that this model performs comparatively to the others we explored (Naive Bayes and support vector machine). Additionally, consistent with previous studies' results, we find that high performance can be achieved by using n-grams and/or Probabilistic Context-Free Grammar production rules as features.

2 Related Work

Early attempts to solve the deception detection problem suffered from a lack of gold-standard data reliably labeled as truthful or deceptive. For example, Jindal and Liu [2008] used a dataset consisting entirely of reviews scraped from Amazon and manually labeled reviews as deceptive if they were duplicates of (or nearly identical to) other

reviews. However, this method only identifies a subset of deceptive reviews. Mihalcea and Strapparava [2009] solicited both truthful and deceptive opinions from Amazon Mechanical Turk, but these opinions were on abstract topics and unrelated to product reviews. Recent work has produced more robust datasets, including those developed by Ott and colleagues [2011,2013] which are discussed in more detail in the following section of this paper; and by Salvetti [2012], which includes reviews in different domains and across different types of deception. Ott and colleagues' [2011] original dataset has also been used in several subsequent studies, including by Feng et al. [2012], Feng and Hirst [2013] and Zhang et al. [2016], and amended to include domain-knowledgeable reviewers by Li et al. [2014].

Many recent studies use a support vector machine (SVM) [Mihalcea & Strapparava, 2009; Ott et al., 2011; Feng et al., 2012; Ott et al., 2013; Feng & Hirst, 2013; Li et al., 2014; Zhang et al., 2016] and/or a Naive Bayes model [Mihalcea & Strapparava, 2009; Ott et al., 2011; Salvetti, 2012] to classify deceptive and truthful reviews. Both models consistently return high accuracies, depending on the features with which they are used. Here, we give some examples of recent feature sets used along with their performance to motivate those used in the current study.

N-grams are frequently used as features in deception detection and tend to produce high accuracies even when used in isolation. For example, Ott et al. [2011] achieve an accuracy of 89.6% on positive reviews only using bigrams alone with an SVM model. This accuracy is only slightly lower than the 89.8% they achieve using bigrams in combination with LIWC features. Li et al. [2014] find that unigrams alone perform best (out of unigrams, LIWC features, and part-of-speech tags) at distinguishing between reviews from actual customers, Mechanical Turk workers, and employees, achieving an accuracy of around 65%.

Another promising source of features comes from Probabilistic Context-Free Grammar (PCFG) production rules. Using Ott and colleagues' [2011] dataset, Feng et al. [2012] reach 91.2% accuracy on an SVM classifier with features consisting of unigrams and lexicalized production rules including grandparent nodes. (We explain these details of PCFG rule-based features in section 4.2 of this paper.) Feng and Hirst [2013] reproduce their approach on the same dataset with 90.1% accuracy. They then supplement the approach with "product profiles" that compare details of a review to actual details about the product, achieving a slightly higher accuracy of 91.3%. Finally, using a combination of PCFG rules and a "co-training" approach, Zhang et al. [2016] achieve an accuracy of almost 95% on the Ott et al. [2011] dataset. Their co-training algorithm, called CoSpa, combines an SVM-based classifier using lexical features with another using PCFG rules; the co-training approach, while unexplored in the current study, suggests a potentially fruitful direction of research.

Additionally, while most recent work has considered classification within a single domain, some studies have examined cross-domain classification. Li et al. [2014] cultivate a dataset consisting of reviews of hotels, restaurants, and doctors and find that models trained in one domain and tested in another perform poorly compared to intra-domain classification, achieving F1 scores no higher than 79%. Salvetti [2012] also reports difficulty with cross-domain classification, achieving a highest accuracy of 63.6% with a Naive Bayes model on a dataset that combines hotel and electronics reviews.

Finally, LIWC features, while shown by Ott et al. [2011] and Li et al. [2014] to be ineffective compared to other features, can be useful for post-hoc analyses of the sorted data. For instance, Mihalcea and Strapparava [2009] identify classes of words from LIWC that are more frequent in either truthful or deceptive descriptions. They find that the top 5 categories for deception are METAPH (metaphysical), YOU, OTHER (other people), HUMANS, and CERTAIN. The top 5 categories for truthfulness are OPTIM (optimistic), I, FRIENDS, SELF, and INSIGHT (words like believe, think, etc.) The authors hypothesize that "truth-related words" from the CERTAIN category might be used more often in deception to overcompensate for the lie. While this study analyzes statements of opinions about people and abstract concepts, the analysis may also inform deception detection in product reviews.

3 Dataset

For their earlier study, Ott et al. [2011] collected 400 positive (5-star) reviews from the travel website TripAdvisor, 20 for each of 20 popular

	Truthful	Deceptive
Positive	We stay at Hilton for 4 nights last march. It was a pleasant stay. We got a large room with 2 double beds and 2 bathrooms, The TV was Ok, a 27' CRT Flat Screen. The coincierge was very friendly when we need. The room was very cleaned when we arrived, we ordered some pizzas from room service and the pizza was Ok also.The main Hall is beautiful. The breakfast is charged, 20 dollars, kinda expensive. The internet access (WiFi) is charged, 13 dollars/day. Pros: Low rate price, huge rooms, close to attractions at Loop, close to metro station. Cons: Expensive breakfast, Internet access charged. Tip: When leaving the building, always use the Michigan Av exit. Its a great view.	My husband and I satayed for two nights at the Hilton Chicago,and enjoyed every minute of it! The bedrooms are immaculate,and the linnens are very soft. We also appreciated the free wifi,as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious,and I loved the smell of the shampoo they provided-not like most hotel shampoos. Their service was amazing,and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.
Negative	My $200 Gucci sunglasses were stolen out of my bag on the 16th. I filed a report with the hotel security and am anxious to hear back from them. This was such a disappointment, as we liked the hotel and were having a great time in Chicago. Our room was really nice, with 2 bathrooms. We had 2 double beds and a comfortable hideaway bed. We had a great view of the lake and park. The hotel charged us $25 to check in early (10am).	We stayed at the Schicago Hilton for 4 days and 3 nights for a conference. I have to say, normally I am very easy going about amenities, cleanliness, and the like...however our experience at the Hilton was so awful I am taking the time to actually write this review. Truly, DO NOT stay at this hotel. When we arrived in our room, it was clear that the carpet hadn't been vacuumed. I figuered, "okay, it's just the carpet." Until I saw the bathroom! Although the bathroom had all the superficial indicators of housekeeping having recently cleaned (i.e., a paper band across the toilet, paper caps on the drinking glasses, etc., it was clear that no ACTUAL cleaning took place. . . . [review continues]

Table 1: Four reviews of the same hotel from the different quadrants of the dataset

Chicago hotels; these reviews were chosen with certain criteria (i.e. greater than 150 characters and by users who had previously written reviews) such that the authors were confident in considering them a gold set of truthful reviews. They then solicited 400 fake positive reviews of the same hotels from Amazon Mechanical Turk to serve as a gold set of deceptive reviews; Turkers were asked to imagine they worked in the hotel's marketing department and had been asked to portray their hotel in a positive, realistic light.

For Ott et al. [2013], the researchers supplemented their dataset with 400 truthful and 400 deceptive negative reviews of the same hotels.The truthful reviews were 1 or 2 star reviews gathered from the travel websites Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, and Yelp; and the deceptive reviews were solicited from Amazon Mechanical Turk with similar instrucitons, only this time the Turkers were asked to imagine writing for a marketing department about a competitor's hotel.

In general and on this dataset in particular, humans perform poorly at deception detection. In classifying truth and deception from the positive data, Ott et al. [2011] found that two out of three human judges could not perform significantly better than chance at detecting deceptive positive reviews. Moreover, judge's had a bias toward classifying reviews as truthful and did not have high agreement between them. With the negative reviews, Ott et al. [2013] indicated that while two out of three human judges did detect negative deceptive reviews better than chance at a statistically significant rate, even the best human judge only achieved accuracy of 65%.

Finally, we'll note that the data, as made available by Ott et al. [2013] has been separated into 5 predefined folds; due to the relatively small size of the dataset, Ott et al. [2011] and [2013], as well as Feng et al. [2012] use 5-fold cross validation to evaluate their models. According to Ott et al.'s folds, all reviews of each hotel are within a single fold, so models are always tested on separate reviews from those they are trained on. We follow this schema to replicate their work as closely as possible.

4 Feature Encoding

4.1 Shallow syntax

Given the effectiveness on bag of word features in Ott et al. [2011], Ott et al. [2013], and subsequent studies, we used unigram and bigram term-frequency features from reviews in the dataset. Ngrams were extracted directly with no transformation (e.g. stemming, stop word removal) other than punctuation removal and lowercasing.

4.2 Deep syntax

Production rules based on Probabilistic Context-Free Grammar parse trees have proved effective in several previous studies (i.e. Feng et al. [2012] and Zhang et al. [2016]). In Feng et al. [2012] (used the Ott 2011 dataset of only positive reviews), the most effective PCFG rule-based features were lexicalized production rules combined with the grandparent node. For example, in the parse tree in Figure 1 (taken from Feng et al. [2012]), features would include VP_2 ^ $NP_2 \rightarrow NP_3$, NP_4 ^ PRP $\rightarrow$ you, etc. where the element before the

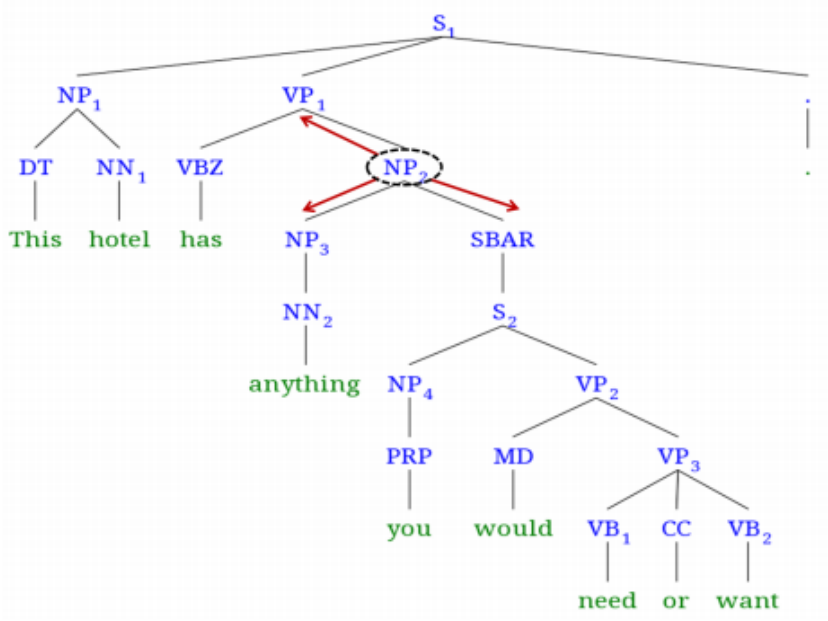

Figure 1: Example PCFG Parse Tree

caret represents the grandparent node. We used the Stanford Parser to parse all sentences in all reviews and then constructed lexicalized production rule features (including grandparent nodes) from each ensuing parse trees.

5 Model

We trained several classifiers on our different feature sets to compare their effectiveness. For each classifier, we trained and evaluated models on just the positive reviews, just the negative reviews, and on the entire set. We used the same 5-fold stratified cross-validation procedure as Ott et al. [2011], [2013] and Feng et al. [2012]. Under this procedure, on each cross-validation iteration we train our model on all reviews in the chosen sentiment category (positive/negative/all) for 16 hotels, and test our model on all reviews in the chosen sentiment category for the remaining 4 hotels.

5.1 Naive Bayes

The Naive Bayes model, which assumes that each feature x is conditionally independent from the others, given a label y of truthful or deceptive, predicts the label for a given example as

$$\arg\max_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Since we always trained with equal numbers of truthful and deceptive examples, the prior $P(y)$ could be ignored; the likelihoods $P(x_i|y)$ of the features were calculated through feature counts with Laplace smoothing. This was a promising model from Ott et al. [2011], and the assumption of conditional independence seems relatively reasonable to use if we assume there exist particular words or syntactic structures that are hallmarks of those who intend to deceive.

5.2 Support Vector Machine (SVM)

Support vector machines attempt to separate the two classes of training data in their feature space by as much of a margin as possible. We followed in the footsteps of Ott et al. [2011] in using a linear kernel for ease of post-hoc analysis of weights; given the high-dimensionality of our features, a linear decision function seems a reasonable way to separate our classes. We used the fast liblinear library via the scikit-learn library to implement our classifier, and we followed Ott et al. [2011]'s and Feng et al. [2012] of using nested cross validation to optimize the C parameter for each step of the outer cross validation's training; we did a grid-search over parameters 10^i for i from -10 to 1.

5.3 Multi-layer Perceptron

We tried training this shallow neural network (just 1 hidden layer, with size ranging from 1 to 100 neurons) with our full set of features (ngrams and PCFG parse rule), because we were interested in seeing if a neural network like this could capture some non-linearity. Given more time, we would be interested in training more complex models (e.g. Recurrent Neural Networks) that could capture the semantics of reviews in subtler ways, but this provided an experiment to see if neural network techniques could be applied to the problem.

6 Results

Results are reported in Tables 2, 3, and 4 for the Naive Bayes, SVM, and Multi-layer Perceptron models, respectively. For each row, we report precision, accuracy, and F1-score, all of which are averaged across truthful and deceptive for each result. Naive Bayes and SVM were run using several different combinations of features, while due to its slow runtime we only tested the multi-layer perceptron model using our most fruitful feature combination (Unigrams, Bigrams, and Deep Syntax).

In most cases, unigrams and PCFG (deep syntax) features did reasonably well on their own. Since our PCFG rule features include leaf node rules (i.e. NP_4 ^ PRP $\rightarrow$ you), we would expect PCFG rule features to perform at least as well as unigrams since leaf node rules are essentially unigram features. Adding in bigrams generally provided a 1% boost in F-1 scores, and combining unigrams, bigrams, and PCFG rule features typically achieved the best results.

Consistent with Ott et al. [2013], we achieved higher accuracy on positive reviews than negative reviews. As in Feng et al. [2012] and Zhang et al. [2016], we achieved the best general results using unigrams, bigrams, and PCFG lexicalized production rules combined with the grandparent node. With Naive Bayes, we were able to achieve accuracy very close to Feng's 91% on the positive set using the same types of features. Our model did slightly better than Ott et al.'s [2013] unigram and bigram-based model on positive-sentiment reviews (90% using all features and Naive Bayes vs 89.3%) but slightly worse on negative-sentiment reviews (85% using all features and an SVM model vs. Ott's 86%). However, our model slightly outperformed Ott's model when trained and tested on positive and negative features combined (88% on Naive Bayes vs. Ott's 87%).

7 Discussion

7.1 Model Comparison

In the end, our Naive Bayes, SVM, and Multi-layer Perceptron models performed at similar levels, with Naive Bayes slightly outperforming the others on the positive and combined datasets. These results reinforce the effectiveness of Naive Bayes and SVM models for deception detection demonstrated in previous studies such as Ott et al. (2011), Feng et al. (2012), and Ott et al. (2013).

There is little to no published data about the effectiveness of neural networks for deception detection, so our preliminary results with the multi-layer perceptron model indicate that simple neural networks can perform with equivalent accuracy to SVM models (at least for the negative and combined datasets). Given these results, more sophisticated recurrent neural networks might be able to capture more latent syntatic differences between truthful and dishonest reviews, providing an interesting future direction for work with deception detection.

7.2 Feature Insights

The feature weights of Naive Bayes, our most accurate model overall, give some hints as to general characteristics that may help to identify deceptive reviews or even deception in general. An examination of the highest weighted shallow syntax features provides a few insights: First, mention of major life events such as weddings and anniversaries (10th anniversary, 10th wedding, 13th anniversary, 13th wedding) came up frequently, indicating that fabricated reviews are more likely to obtain references to such occasions. Second, mention of prices (500 dollars, 600 night, etc.) were strong predictors of deceptive reviews, which makes sense because it would be easier for someone to obtain price data even if they haven't stayed at the hotel in question. Third, highly emotive adjectives (great, terrific, horrifying, unsurpassable, unparalleled) were also strongly associated with deceptive reviews, corroborating Ott el al.'s [2013] finding. This seemed to be true for both positive and negative adjectives.

An examination of the highest-weight deep syntax features potentially sheds some light on

	Positive			Negative			Combined		
	P	R	F	P	R	F	P	R	F
Unigrams	89	89	88	84	84	**84**	86	86	86
Unigrams + Bigrams	90	89	89	83	81	81	87	87	87
Deep Syntax Features	89	89	88	81	80	79	85	84	84
Unigrams, Bigrams, and Deep Syntax	90	90	**90**	84	82	81	89	88	**88**

Table 2: Naive Bayes Results (F1-scores)

	Positive			Negative			Combined		
	P	R	F	P	R	F	P	R	F
Unigrams	84	84	84	83	83	83	85	85	85
Unigrams + Bigrams	87	86	86	84	84	84	85	84	84
Deep Syntax Features	87	87	87	85	85	85	86	86	86
Unigrams, Bigrams, and Deep Syntax	88	88	**88**	85	85	**85**	87	87	**87**

Table 3: SVM Results (F1-scores)

	Positive			Negative			Combined		
Hidden Layer Size	P	R	F	P	R	F	P	R	F
1	75	72	72	70	70	70	74	70	69
5	85	85	85	79	77	77	86	86	86
10	84	84	84	84	83	83	87	87	**87**
50	88	88	**87**	85	85	**85**	86	86	86
100	86	86	86	85	85	85	87	87	87

Table 4: Multi-layer Perceptron Results (F1-scores using Unigrams, Bigrams, and Deep Syntax Features)

phrasal patterns that help discriminate between truthful and deceptive reviews. As expected given the success of unigrams, many of the highest-weighted PCFG rule features were terminal node production rules (i.e. NP ^ JJ $\rightarrow$ premier, ADJP ^ JJ $\rightarrow$ top-notch) which essentially functioned as unigram features.

While such terminal node features, much like term-frequency features, are highly context-dependent, non-lexicalized production rule features may provide more generalizable insight into phrasal patterns that distinguish deceptive writing from truthful writing. In this domain, we found that adjective-rich phrasal expansions (i.e. ADJP ^ ADJP $\rightarrow$ ADJP ADJP, PP ^ NP $\rightarrow$ NP ADJP VP) were highly weighted. This suggests that deceptive reviewers not only use more emotive adjectives but also include more adjective-rich phrasal content in general.

In addition, consistent with the results of Zhang et al. [2016], past-tense production rules (e.g. X ^ VBD $\rightarrow$ was, VP ^ VP $\rightarrow$ hadnt, SQ ^ VBD $\rightarrow$ were) frequently appeared among the highest-weighted features.

7.3 Error Analysis

For the positive, negative, and combined datasets, Naive Bayes with all features had low precision but high recall for deceptive reviews and high precision but low recall for truthful reviews. For example, in Naive Bayes with all features, for truthful reviews P=84% and R=94% but for deceptive reviews P=93% and R=82%. Thus, contrary to the truth bias exhibited by human judges, our model tended to misclassify true reviews as deceptive.

Upon examining truthful reviews that were misclassified as deceptive, we observe a couple features that may be helpful in improving classification. First, several of these reviews referred to verifiable people and events that would be difficult to fabricate if one had not visited the hotel. For example, one reviewer claims "We where [sic] woken up at 6:00 in the morning with our bathroom flooded, this water was brown and all the clothing and our belongings left in the bathroom were dirty and sopping wet." Another says "Vanessa at the

front desk, went out her way to help me - storing some things in a safe for me before I could book into my room." A third says "I checked into this hotel, Rm 1760 on 11/13/2010. Next day, I woke up with bed bug bites, numerous, on my arm and back."

Although the inclusion of hand-built features related to the inclusion of specific details like room number, staff names, etc. would potentially improve classifier accuracy, creating these features requires context-specific knowledge for each domain a classifier is trained in. Thus, in an attempt to create generalizable strategy for a classifier that can be trained on any domain with the same general feature sets, this would not be the best approach.

Another interesting trend in misclassified truthful reviews was the inclusion of references to other reviewers. For example, one reviewer wrote "It seems as though I had the same experience as Judy L. I read her review and thought that's exactly what happened to me!" To some extent, this may be a factor of how deceptive reviews were solicited, since Mechanical Turk reviews were written in isolation while genuine deceptive reviews could easily refer to other reviewers on the same website. However, if it is the case that truthful reviews are more likely to refer to other reviewers, it would be fairly easy to include number of references to other reviewers as a feature, and this would be more generalizable across domains than the hand-built context-specific features described above.

7.4 Dataset Reliability

Although the Ott dataset has been used extensively in deception detection studies, there are several considerations to be aware of before attempting to draw conclusions about the generalizability of our models. Salvetti [2012] brings up some potential issues with MTurk deceptive hotel reviews, question whether Turkers can really lie as convincingly as real spammers. Although the Turkers were United States residents, they had not necessarily even been to Chicago, much less any of the hotels they wrote about. On the other hand, spammers are more likely to have more domain knowledge, which they can leverage to write more convincingly deceptive reviews. Due to this potential discrepancy, Salvetti created the BLT-C (Boulder Lies and Truths Corpus), a new dataset of hotel and electronic reviews in which all reviews are written by Turkers and deception categories are more nuanced, with T for true reviews about a known object (i.e. a hotel or electronic device that the reviewer has personally used), F for false reviews about a known object, and D for fabricated reviews about objects with which the reviewer does not have direct experience. All reviews are written by Turkers to eliminate any confounding factors that may arise from having only false reviews written by Turkers. In the future, it may be valuable to test this and other models on the BLT-C to ensure that the results obtained reflect something distinctive about deceptive reviews rather than just something distinctive about reviews written by Turkers.

That being said, the low rates of human accuracy in determining deceptive reviews indicates that the deceptive reviews written by Turkers were at least reasonably convincing, and the fact remains that our model does far better than human judges at picking out deceptive reviews.

7.5 A Social Component

Our work has focused strictly on detecting deception for single reviews in isolation using linguistic features. However, it would likely be useful to also consider the social component of reviews in order to build a truly robust deception detection system. As noted above, our error analysis revealed inclusion of other reviewers' names could serve as a useful feature to help reduce misclassification of truthful reviews. Moreover, many users review a variety of products, so the deception ratings for other reviews by a given user may be helpful in predicting whether or not the review currently being evaluated is deceptive. Moreover, work by Wang et al. [2011] indicates that a review graph connecting reviewers to stores and each other may reveal networks of spammers that even the most sophisticated textual analysis model could not identity. A social graph model and textual analysis model could be extremely symbiotic, which the textual analysis model providing likelihood of deception for individual reviews and the social graph model providing likelihood of deception for individual *reviewers*, which could then be used as a feature in the textual analysis model. A deception detection system employing both of these approaches could thus be extremely successful.

8 Conclusion

Our results confirm the effectiveness of both n-gram and PCFG rule-based features, indicating that high accuracy can be achieved from unigrams alone and deep syntax features provide a small but consistent boost. We also demonstrated that neural networks can provide a promising direction for further development of deception detection models, as our shallow multi-layer perceptron model achieved accuracy approximately on par with our Naive Bayes and SVM models. This indicates that applying recurrent neural networks to deception detection could be a promising direction for future work.

Furthermore, in order to establish the robustness of the model, it may be useful to test on the BLT-C and other datasets. Finally, a combined approach that employs both linguistic analysis and a social graph model seems to be a encouraging future direction for creating a system to more accurately detect deceptive product reviews.

References

Feng, S., Banerjee, R., and Choi, Y. 2012. Syntactic stylometry for deception detection. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. 171-175.

Feng, V.W., and Hirst, G. 2013. Detecting deceptive opinions with profile compatibility. *International Joint Conference on Natural Language Processing*. 338-346.

Jindal, N., and Liu, B. 2008. Opinion spam and analysis. *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 219-230.

Li, J., Ott, M., Cardie, C., and Hovy, E. 2014. Towards a general rule for identifying deceptive opinion spam. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 1566-1576.

Mihalcea, R., and Strapparava, C. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. 309-312.

Ott, M., Cardie, C., and Hancock, J. T. 2013. Negative deceptive opinion spam. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Short Papers*. 497-501.

Ott, M., Cardie, C., and Hancock, J. T. 2011. Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. 309-319. Association for Computational Linguistics.

Salvetti, Franco 2012. *Detecting Deception in Text: A Corpus-driven Approach*. University of Colorado at Boulder, Boulder, CO.

Wang, G., Xie, S., Liu, B., and Yu, P. S. 2011. Review graph based online store review spammer detection. *Data mining (icdm), 2011 ieee 11th international conference*. 1242-1247.

Zhang, W., Bu, C., Yoshida, T., and Zhang, S. 2016. CoSpa: A co-training approach for spam review identification with Support Vector Machine. *Information* 7(1), 12.