Detecting deceptive opinions with profile compatibility

Vanessa Wei Feng
University of Toronto
Toronto, ON, M5S 3G4, Canada
weifeng@cs.toronto.edu

Graeme Hirst
University of Toronto
Toronto, ON, M5S 3G4, Canada
gh@cs.toronto.edu

Abstract

We propose using profile compatibility to differentiate genuine and fake product reviews. For each product, a *collective* profile is derived from a separate collection of reviews. Such a profile contains a number of aspects of the product, together with their descriptions. For a given unseen review about the same product, we build a *test* profile using the same approach. We then perform a bidirectional alignment between the *test* and the *collective* profile, to compute a list of aspect-wise compatible features. We adopt Ott et al. (2011)'s op_spam_v1.3 dataset for identifying *truthful* vs. *deceptive* reviews. We extend the recently proposed N-GRAM+SYN model of Feng et al. (2012a) by incorporating profile compatibility features, showing such an addition significantly improves upon their state-of-art classification performance.

1 Introduction

With the rapid development of e-commerce and the increasing popularity of various product review websites, people are more and more used to making purchase decisions based on the reported experience of other customers. A product rated positively by its previous users is able to attract potential new customers, while a poorly rated product is certainly not a good option for most new customers. Given this influential power of product reviews, there comes a huge potential for *deceptive opinion spam* to distort the true evaluation of a product. The promoters of a product may post false complimentary reviews, and competitors may post false derogatory reviews.

Although the task of detecting *deceptive opinion spam* can be formulated as a traditional binary classification problem with two classes, *deceptive* and *truthful*, where supervised learning can be applied, it is essentially very challenging. Its major difficulty arises from the lack of reliably labeled data, because it is extremely difficult for humans to identify through reading (see discussion in Section 4). Therefore, much previous work on detecting deceptive opinions usually relies on some meta-information, such as the IP address of the reviewer or the average rating of the product, rather than the actual content of the review Liu (2012). However, thanks to the release of the op_spam_v1.3 dataset by Ott et al. (2011), a gold-standard dataset composed of 400 truthful and 400 deceptive hotel reviews (see Section 3), we are now at an appropriate stage for conducting supervised learning and reliable evaluation of the task.

In their work, Ott et al. proposed to use *n*-grams as features, and trained an SVM classifier to classify whether a given review is deceptive or truthful, achieving nearly 90% accuracy. More recently, Feng et al. (2012a) extended Ott et al.'s *n*-gram feature set by incorporating deep syntax features, i.e., syntactic production rules derived from Probabilistic Context Free Grammar (PCFG) parse trees, and obtained 91.2% accuracy on the same dataset.

While both Ott et al. and Feng et al.'s cues of deception come purely from the surface realization in the reviews, we propose the use of additional signals of truthfulness by characterizing the degree of *compatibility* between the personal experience described in a test review and a product profile derived from a collection of reference reviews about the same product. Our intuition comes from the hypothesis that since the writer of a deceptive review usually does not have any actual experience with that product, the resulting review might contain some contradictions with facts about the product, or the writer might fail to mention those aspects of the product that are commonly mentioned in truthful reviews. Conversely, a writer of a truthful review is more likely to make similar comments about some particular aspects of the product as other truthful reviewers. In other words, we postulate that by aligning a the profile of a product and the description of the writer's personal experience, some useful clues can be revealed to help identify possible deception.

In line with the intuition above, we want to capture two types of compatibility: (1) Compatibility with the existence of some *distinct* aspect of the product, such as the mention of the famous art museum nearby the hotel; (2) Compatibility with the description of some *general* aspect of the product, such as commenting that the breakfast at the hotel is charged for.

We show that, by incorporating our computed profile alignment features with Feng et al.'s *n*-grams and deep syntax features, we significantly improve on the state-of-art performance presented by their N-GRAM+SYN model.

2 Related work

Before the existence of gold-standard datasets which include both truthful and deceptive product reviews, several attempts were made to detect deceptive opinions.

With regard to unsupervised approaches, previous work was primarily based on various patterns of atypical behaviors. Lim et al. (2010) proposed several behavior models to identify unusual reviewer patterns; e.g., spammers may contribute a fake review soon after product launch to maximally affect subsequent reviewers. Each of these individual models assigns a numeric score indicating the extent to which the reviewer has engaged in apparent spam behaviors, and these scores are then combined to product a final spam score. Jindal et al. (2010) employed data-mining techniques to discover unexpected class association rules; e.g., reviewers who give all high ratings to products of a brand while most other reviews are generally negative about the brand. Wu et al. (2010) proposed to use the distortion of popularity rankings as an indicator of opinion spamming, in the sense that deleting a set of random reviews should not overly disrupt the popularity ranking of a list of entities, while deleting fake reviews should significantly distort the overall rankings, under the assumption that deceptive reviews usually express a sentiment at odds with legitimate reviews. Feng et al. (2012b) postulated that for a given domain, there exists a set of representative distributions of review rating scores, and fake reviews written by hired spammers will distort such natural distributions. Experimenting with a range of pseudo-gold-standard datasets, they provided quantitative insights into the characteristics of natural distributions of opinions in various product review domains.

With respect to supervised approaches, Jindal and Liu (2008) first conducted a tentative study of detecting deceptive opinions. In the absence of gold-standard datasets, they trained models using features from the review texts, as well as from meta-information on the reviewer and the product, to distinguish between *duplicate* reviews (regarded as deceptive), i.e., reviews whose major content appears more than once in the corpus, and *non-duplicate* reviews (regarded as truthful). However, these (near-)duplicate reviews are often not sophisticated in their composition, and therefore are relatively easy to identify, even by off-the-shelf plagiarism detection software (Ott et al., 2011).

Yoo and Gretzel (2009) constructed a small gold-standard dataset with 40 truthful and 42 deceptive hotel reviews, and manually inspected the statistical differences of psychologically relevant linguistic features for truthful and deceptive reviews.

Ott et al. (2011) released the op_spam_v1.3 dataset (see Section 3), a much larger dataset with 400 truthful and 400 deceptive hotel reviews with *positive* comments, which allows the application of machine learning techniques for training models to automatically detect deceptive opinions. In their work, Ott et al. focused on the textual content of the reviews, and approached the task of detecting deceptive opinions by finding any *stretch of imagination* in the text — they treated fake reviews as imaginative writing, and used standard computational approaches of *genre identification*, *psycholinguistic deception detection*, and *text categorization* to identify them. Among the set of features explored in their classification experiments, they discovered that the features traditionally employed in either psychological studies of deception or genre identification were both significantly outperformed by a much simpler N-GRAM model with *n*-grams only as features, which achieved nearly 90% accuracy. Later, Feng et al. (2012a) proposed a strengthened model, N-GRAM+SYN, by incorporating syntactic production rules derived from Probabilistic Context Free Grammar (PCFG) parse trees. They obtained 91.2% accuracy on the op_spam_v1.3 dataset, which is a 14% error reduction.

More recently, Ott et al. released a new version, the op_spam_v1.4 dataset, with gold-standard *negative* reviews included as well (Ott et al., 2013), which offers an opportunity to more extensively study the problem of detecting deceptive reviews. However, the work described in this paper is focused on *positive* review spam, and based directly on the work of Feng et al. We extend their N-GRAM+SYN model by incorporating potential signals of truthfulness, derived from aligning the description in a test review with the product profile constructed from a large collection of reference reviews about the same product. Somewhat orthogonal to Feng et al.'s *n*-grams and deep syntax features, which are focused on the surface realization of a given product review, our alignment features are able to employ useful information from the product itself.

3 The op_spam_v1.3 Dataset

Due to the difficulty for humans to identify deceptive opinions, traditional approaches to constructing annotated corpora by recruiting human judges to label a given set of texts does not apply to the task of deception detection. Consequently, crowdsourcing services such as Amazon Mechanical Turk[1] (AMT) have been adopted as a better solution (Ott et al., 2011; Rubin and Vashchilko, 2012). By asking paid subjects on AMT to compose deceptive and/or truthful texts, corpora with reliable labels can be constructed.

In this work, we use Ott et al.'s op_spam_v1.3 dataset[2], which contains **positive** reviews for the 20 most-rated hotels on TripAdvisor[3] (TA) in Chicago. For each hotel, Ott et al. selected 20 deceptive reviews from submissions on AMT, and 20 truthful reviews from 5-star reviews on TA, resulting in 800 reviews in total for 20 hotels. The average length of deceptive reviews is 115.75 words, while the truthful reviews are chosen to have roughly the same average length.

[1]`http://mturk.com.`
[2]`http://www.cs.cornell.edu/~myleott`
[3]`http://tripadvisor.com.`

4 Difficulty of the Task

It has been shown that humans can differentiate truthful reviews from deceptive ones with only a modest accuracy. With respect to the op_spam_v1.3 dataset used in our work, human judges were only able to achieve 60% accuracy, and the inter-annotator agreement was low: 0.11 computed by Fleiss's kappa (Ott et al., 2011). Since as humans, we do not have a tangible intuition of what signals deception and what characterizes truthfulness, we have great difficulty designing features for automatic identifying deceptive opinions.

In addition, there is difficulty with regard to the construction of gold-standard datasets for the task of deception detection. On one hand, although crowdsourcing provides a relatively cheap and reliable approach, gathering deceptive data is still nevertheless laborious. On the other hand, since for truthful reviews, we are able to select a small subset for which we have high confidence only by using a combination of various heuristics, the constructed dataset is inherently biased. Therefore, given the limited number and size of currently available gold-standard datasets, in the process of developing more sophisticated models involving a larger number of features, we may inevitably face the issue of overfitting on a relatively small set of data.

5 Methodology

5.1 Baseline features

We adopt Feng et al.'s N-GRAM+SYN model as the baseline system, which employs two distinct sets of features.

N-GRAM features: As shown by Ott et al., the bag-of-words model is effective for identifying deceptive reviews. However, the optimal choice of *n* is not consistent in Ott et al.'s original implementation and Feng et al.'s strengthened model: Ott et al. used the union of unigrams and bigrams, while Feng et al. obtained their best performance using unigrams alone, together with deep syntax features. Therefore, for a fair comparison, we consider using unigrams, bigrams, and the union of both, and choose the best combination with deep syntax features as our baseline system.

SYN features: Following Feng et al., deep syntax features are encoded as production rules derived from PCFG parse trees. These production rules include lexicalized ones, i.e., POS-tag-to-word rules, and are combined with the grandparent node.

All baseline features are encoded as their tf-idf weights, with features appearing only once in the dataset eliminated.

5.2 Product profile construction

5.2.1 Aspects and compatibility

As introduced in Section 1, we postulate that by aligning the profile of a product with the description of the writer's personal experience, we can characterize the degree of compatibility between them. Such alignment features could serve as useful signals to differentiate between deception and truthfulness.

In particular, we define compatibility for two types of aspects of a product — *distinct* aspects and *general* aspects. Distinct aspects are special features of the product. For hotels, those distinct aspects are usually realized as proper noun phrases in the text, typically the names of the landmarks nearby the hotel, including museums, parks, restaurants, bars, and shopping malls. On the other hand, general aspects are common features which typically appear in any product of this particular kind. For hotels, **location**, **service**, and **breakfast** are typical general aspects. The method we use to identify and extract distinct and general aspects from reviews is described in Section 5.2.4.

Compatibility for distinct and general aspects is defined as follows:

1. Compatibility with the **existence** of some *distinct* aspect of the product, e.g., truthful reviews of this Chicago hotel often mention the famous nearby Field Museum, and the test review also mentions this museum.

2. Compatibility with the **description** of some *general* aspect of the product. For example, breakfast at most Chicago hotels is complimentary; however, the breakfast at this hotel is charged for, and the test review describes it as "the breakfast is kinda expensive".

5.2.2 Definition of product profile

A product's profile P is composed of a set of its aspects $A = \{a_1, \dots, a_N\}$, where each a_i is an individual aspect of the product. Each aspect a_i is associated with a description $D_i = \{p_1 : w_1, \dots, p_m : w_m\}$, in which each element is a unique word p_j to describe the aspect, along with the weight w_j assigned to that word.

Aspect	**Description**
Bathroom	$\{$*clean* : 3.0, *comfortable* : 3.0, *pleasant* : 5.0, *high-end* : 1.0, *European* : 1.0$\}$
Room	$\{$*wonderful* : 4.0, *deluxe* : 2.0, *huge* : 2.0$\}$
Service	$\{$*average* : 2.0, *slow* : 2.0$\}$
Michigan Ave.	$\{$*existence* : 5.0$\}$

Table 1: An example fragment of the profile of a hotel.

Distinct aspects, $a_{i,d}$, and general aspects, $a_{i,g}$, are treated differently when it comes to their descriptions. For a distinct aspect, its description can contain only one word, *existence*, because we only care about whether this aspect is mentioned in the text, not the particular words used to describe this aspect. In fact, most of these distinct aspects do not occur with any associated adjectives or adverbs.

An example fragment of a hotel's profile is shown in Table 1, in which the last aspect **Michigan Ave.** is a distinct aspect, and thus only one word, *existence*, appears in its description.

5.2.3 Data for profile construction

To establish a profile P for each target hotel in the op_spam_v1.3 dataset, we first gather all its reviews on TripAdvisor, from which we choose up to 200 reviews, subject to the following criteria:

1. It is not present in the op_spam_v1.3 dataset.

2. Its rating is five stars (the highest).

3. Its language is English.

4. It contains at least 150 characters.

5. The author has written at least 10 reviews.

6. The author has written reviews for at least 5 different hotels.

7. The author has received at least 5 helpfulness votes from other users.

If there are more than 200 reviews satisfying the above criteria, we choose the 200 with the highest number of helpfulness votes received by their authors. Most of the above criteria are consistent

Reviews			
	Min	*Max*	*Mean*
Reviews per hotel	44	200	160.6
Characters	150	8,995	848.3
Authors			
	Min	*Max*	*Mean*
Total reviews	10	506	37.8
Reviewed hotels	5	278	17.8
Helpfulness votes	5	2,280	33.3

Table 2: The statistics of the collection of reviews used in constructing profiles and their distinct authors.

with the original setup of collecting the truthful reviews in the op_spam_v1.3 dataset. However, we add the last three strong criteria to ensure the quality of the collected reviews.

Table 2 lists the statistics of the reviews used in our profile construction. The statistics are categorized into (1) the information about the reviews themselves, including the total number of qualified reviews for each hotel, and the length (in characters) of each review; and (2) the information about the distinct authors of these reviews, including the total number of reviews each author has written on TripAdvisor, the total number of hotels each author has reviewed, and the total number of helpfulness votes that each author has received from other users.

5.2.4 Aspect extraction

Raw aspect extraction For each particular hotel h, from the set of its corresponding reviews R_h, we extract the aspects $A_h = \{a_{1,t_1,h}, \ldots, a_{M,t_M,h}\}$ (e.g., **room**, **service**, and **Michigan Ave.** in Table 1) to be included into h's profile P_h, where t_i is the type, *distinct* or *general*, of each aspect. For the sake of notational clarity, from now on, when talking about a specific hotel h, we will omit the subscript h. We first extract all noun phrases present in R, and use these noun phrases as raw aspects. For each raw aspect extracted, we identify whether it is a *distinct* or *general* aspect: if the aspect is extracted as a proper noun phrase (as tagged by the Stanford parser (Klein and Manning, 2003)), then the aspect is labeled as *distinct*, and *general* otherwise.

Noun phrase extraction is performed solely on the syntactic parse trees. We do not attempt to resolve coreferences for the following two reasons: (1) since most coreference resolution tools are trained on news texts, their performance on product reviews might not be reliable; (2) we observe that authors of these product reviews rarely employ coreference in their writing, and therefore the impact of resolving coreference might be minor after all.

Aspect clustering Due to the flexibility of natural languages, it is common to see people express the same concept through different lexical usage. For example, both "price" and "rate" can refer to the price of the hotel. In addition to synonyms, semantically related words can also be adopted to describe the same aspect of the product, for example, "lighting" and "painting" can both describe the decoration of the hotel.

In order to deal with the use of these (near-) synonyms in text, we perform separate clustering for the sets of distinct and general aspects.

For distinct aspects $\{a_{i,d}\}$, the distance between a pair of aspects is defined by the length of their longest common substring divided by the average length of the two aspects. The motivation for this definition is to bring *distinct* aspects such as **Michigan Ave.** and **Michigan Avenue** into the same cluster.

For general aspects $\{a_{i,g}\}$, the distance between a pair of aspects is defined as their Lin Similarity (Lin, 1998), implemented by NLTK's similarity package (Bird et al., 2009).

For both types of aspects, we perform hierarchical agglomerative clustering with average linkage. Cluster merging is terminated once the distance between all pairs of clusters is greater than 0.3.

Sorting aspects by occurrences After clustering distinct and general aspects separately, we keep track of the total occurrences of each aspect across the entire set R. Then, we let A be the set of the most common N aspects[4]. N is a parameter identical for all hotels, indicating how many aspects are included in each profile. We expect that the choice of N is affected by the particular type of product, and we experiment with several values of N.

[4]N is usually much smaller than M, the total number of raw aspects.

5.2.5 Description extraction

For a hotel h, each of its aspects a_i is then associated with a description

$$D_i = \{p_{i,1} : w_{i,1}, \ldots, p_{i,m} : w_{i,m}\},$$

which has two components: description words $p_{i,j}$ and their weights $w_{i,j}$, for $1 \leq j \leq m$.

Description words The list $\{p_{i,1}, \ldots, p_{i,m}\}$ is composed of the description words used to modify the particular aspect a_i, e.g., $\{$*wonderful*, *deluxe*, and *huge*$\}$ for the aspect **room** in Table 1.

For each *distinct* aspect $a_{i,d}$, as discussed in Section 5.2.2, there can be only one description word, *existence*, indicating the presence of this aspect in the profile.

For each *general* aspect $a_{i,g}$, its description words are automatically extracted from the dependencies in the set of reviews R. From the dependencies output by the Stanford dependency parser (de Marneffe et al., 2006), for a particular aspect a_i, we first locate all relevant dependency relations which have a_i (or the head noun of a_i if a_i is a noun phrase) as the governors (or dependents, depending on the particular dependency types). Then from these relevant dependency relations, we extract the words that appear in the dependent (or governor) slots to be the description words, provided that those words are tagged as adjectives or adverbs.

Weight assignment Each description word $p_{i,j}$ is assigned a weight $w_{i,j}$ to indicate the *reliability* of that particular description word. There are many possibilities for these weight assignment functions. Here, as a preliminary study, for *distinct* aspects $\{a_{i,d}\}$, we simply use the number of occurrences of the aspect in the review collection R as the assigned weight $w_{i,j}$, and for *general* aspects $\{a_{i,d}\}$, we use the number of occurrences of $p_{i,j}$ in R as $w_{i,j}$.

5.3 Computing profile compatibility

Once we establish a profile P for each hotel h from the set of relevant reviews R, then given an unseen test review r about h, we can perform a *bidirectional* alignment with P, and compute a list of compatibility features.

Using the same approach as described in Sections 5.2.4 and 5.2.5, we construct a profile Q from this single review. For the sake of clarity, we call Q the *test profile*, in contrast to P, the *collective profile*, which is constructed from a collection of reviews.

5.3.1 Bidirectional alignment

We perform a *bidirectional* alignment between the test profile P and the collective profile Q. The *direction* of an alignment is defined as the following.

When aligning the test profile Q with the collective profile P ($Q \rightarrow P$), we compute *aspect-wise* compatibility features (to be defined in Section 5.3.2) **for each aspect a_i in** P. Similarly, when aligning the collective profile P with the test profile Q ($P \rightarrow Q$), we compute **for each aspect a_i in** Q.

The $Q \rightarrow P$ alignment captures whether the aspects that most reviewers would describe in their reviews are mentioned in the test review, and whether there exist any similarity or conflicts between the common opinions represented in the collective profile P and the test review.

The $P \rightarrow Q$ alignment captures whether the test review falsely includes some imaginary extra aspects about the product, such as an indoor pool in the hotel, while the collective profile P does not include this aspect.

5.3.2 Aspect-wise compatibility features

For each direction of the bidirectional alignment, we compute a list of aspect-wise compatibility features. Without loss of generality, assume that the direction of the alignment is $Q \rightarrow P$. Thus, for each aspect a_i in P, we include three features to indicate the compatibility between P and Q:

1. m_{a_i}: a boolean value indicating whether a_i is mentioned in Q or not. If a_i does appear in Q, we then compute the following two values, based on a_i's description D_i in P, and its description D'_i in Q.

2. $s_{a_i} \in [0, 1]$: a numeric value indicating the *similarity* between P and Q with respect to a_i. S_{a_i} is computed as the cosine similarity between D_i and D'_i. We treat D_i and D'_i as two vectors following the bag-of-words model, where the description words are the words, and the weights are their corresponding frequencies.

3. $c_{a_i} \in [0, 1]$: a numeric value indicating the *conflicts* between P and Q with respect to a_i. For each description word p_i in D_i, we determine whether D'_i includes its antonym.

Antonyms are determined using the lexical relations defined in WordNet[5]. c_{a_i} is computed as the fraction of description words in D_i that have corresponding antonyms in D'_i.

Clearly, when computing aspect-wise similarity score s_{a_i}, we do not consider any synonyms or semantic similarity between two description words — the similarity is based solely on the use of exact words. Our strategy of exact word matching is a simplification, but since similarity of adjectives and adverbs is not as well-studied as that of nouns (various similarity metrics for nouns are available for WordNet, but none of those work on adjectives and adverbs), clustering description words is not as trivial as clustering aspects[6].

5.4 *C*+N-GRAM+SYN model

To test whether a given review r about the hotel h is deceptive or not, we extend Feng et al.'s N-GRAM+SYN model by incorporating our aspect-wise compatibility features into it. We call this extension the C+N-GRAM+SYN model, which includes the following two categories of features:

1. Alignment compatibility features
   (a) Compatibility features for alignment $Q \rightarrow P$, including m_{a_i}, s_{a_i}, and c_{a_i} for each of the N aspects $\{a_1, \ldots, a_N\}$ with the highest occurrences in P.
   (b) Compatibility features for alignment $P \rightarrow Q$, including $m_{a'_i}$, $s_{a'_i}$, and $c_{a'_i}$ for each of the N aspects $\{a'_1, \ldots, a'_N\}$ with the highest occurrences in Q.
2. Baseline features: 3,009 unigrams, 10,538 bigrams, and 6,571 syntactic production rules, as described in Section 5.1.

The resulting dimension of our full feature set is 20,118 + $2N$, where N ranges from 10 to 100 in our experiments (see Section 6.2). For all experiments, we use SVM^{perf} (Joachims, 2006) to train all the linear SVM classifiers.

6 Results and discussion

6.1 Reimplementing baseline models

We first demonstrate the performance of our reimplemented baseline models. We conduct 5-fold cross-validation experiments over the 800 reviews of 20 hotels, in which each fold contains all reviews from four hotels, such that the learned models are always tested against unseen hotels. Performance is reported by accuracy and the micro-averaged precision, recall, and F-score of retrieving *deceptive* reviews (see Ott et al. (2013) for details). We experiment with five different baseline feature sets: (1) SYN: syntax features, (2) N-GRAM: unigram and bigram features, (3) 1-GRAM+SYN: unigram and syntax features, (4) 2-GRAM+SYN: bigram and syntax features, and (5) N-GRAM+SYN: unigram, bigram and syntax features, where the second is Ott et al.'s model[7], and the third is Feng et al. (2012a)'s model.

The results are shown in Table 3. Our reimplementation of Ott et al.'s model (N-GRAM) obtains 89.6% accuracy, while the accuracy of our reimplemented Feng et al. model (1-GRAM+SYN) is noticeably lower (88.9%). Though its improvement over the N-GRAM model is not significant, the N-GRAM+SYN model performs the best (90.1%) among all baseline models, and thus is chosen to be the baseline in our later experiments.

Features	**Prec**	**Rec**	F_1	**Acc**
SYN	87.2	88.8	88.0	87.9
N-GRAM	89.7	89.5	89.6	89.6
1-GRAM+SYN	88.0	90.0	89.0	88.9
2-GRAM+SYN	87.9	90.5	89.2	89.0
N-GRAM+SYN	*89.2*	*91.3*	*90.2*	*90.1*

Table 3: Results (in %) of our reimplemented baseline models. Performance is reported using 5-fold cross-validation over the op_spam_v1.3 dataset (800 reviews of 20 hotels).

6.2 Comparison with baseline models

We now demonstrate the performance achieved by our profile alignment compatibility features. We experiment with using our profile alignment features in combination with baseline features, i.e., the C+N-GRAM+SYN model. We first study the

[5]http://wordnet.princeton.edu/wordnet/

[6]In fact, we have experimented with using LSA-based word vectors to cluster description words, but the performance is slightly lower than using exact word matching.

[7]Actually, the best performance reported by Ott et al. was obtained using their LIWC + N-GRAM model, which is the combination of N-GRAM features and 80 features output by the Linguistic Inquiry and Word Count (LIWC) software (Pennebaker et al., 2007). However, since the improvement over n-gram features alone was small and insignificant (0.2% absolute improvement in accuracy), and we were not able to successfully reproduce the improvement, the comparison with our model is based on using N-GRAM features alone.

effect of the number of reviews used in constructing collective profiles (see Section 5.2.3). We experiment with using the top 50, 100, 150, and 200 reviews written by authors with the highest helpfulness votes (for some hotels, the total number of available reviews is less than 200). For each set of constructed profiles, we tried different values of N, the number of aspects included in each profile (both collective and test profiles), ranging from 10 to 100, and determine the optimal number by conducting 4-fold cross-validation experiments using 80% of the training data.

The results are shown in Table 4. First, we see that the best overall performance, in terms of F_1 score and accuracy, is achieved using the top 50 or 100 reviews for profile construction. This suggests that using a fairly small amount of reliable data is sufficient to recognize compatibility between collective profiles and test reviews. In fact, using a larger number of reviews is slightly detrimental to the overall performance, since more noise might be included in the process.

Moreover, we see that our best model, achieved using 50 or 100 reviews for profile construction, outperforms the best baseline model, N-GRAM+SYN, on all four metrics, by over 1%, which is a 12.1% error reduction rate. The improvement is confirmed to be statistically significant ($p < .05$) using the Wilcoxon sign-rank test. In addition, we inspect each model's predictions on individual test instances, and discover that our C+N-GRAM+SYN model correctly classifies some instances on which the baseline model makes an error, and at the same time does not make any additional errors that the baseline model does not make.

Finally, with respect to the value of N, which is the number of aspects included in each profile, we discover that smaller values of N, i.e., 10 or 20, consistently give the most satisfying results.

Reviews used	N	**Prec**	**Rec**	F_1	**Acc**
50	10	90.2*	92.5*	**91.4***	**91.3***
100	10	90.0*	**92.8***	**91.4***	**91.3***
150	20	**90.6***	92.0*	91.3*	**91.3***
200	20	90.4*	91.8*	91.1*	91.0*

Table 4: Results (in %) of our profile alignment compatibility models. Performance is reported using 5-fold cross-validation over the op_spam_v1.3 dataset (800 reviews of 20 hotels). The best result of each metric is shown in bold face. All numbers are significantly better (denoted by *) than the baseline in Table 3, using the Wilcoxon sign-rank test ($p < .05$).

7 Conclusions and perspectives

In this paper we proposed the use of profile alignment compatibility as an indicator of truthfulness in product reviews. We defined two types of compatibility between product profiles, and designed a methodology to tackle them by extracting aspects and associated descriptions from reviews. We adopted Ott et al.'s op_spam_v1.3 dataset of hotel reviews, and improved the N-GRAM+SYN model of Feng et al. (2012a). Our approach was shown to significantly improve the performance of identifying deceptive reviews.

In future work, it is critical to know how well our methodology of extracting aspects and their descriptions from hotel reviews generalizes on other kinds of reviews. We suspect that the approach should work equally well on other domains, as long as the aspects are realized mainly by noun phrases, especially that distinct aspects are realized by proper noun phrases.

Another particularly interesting direction is to explore how our C+N-GRAM+SYN model performs on identifying fake *negative* reviews, as recently released by Ott et al. (2013), rather than the *positive* reviews used in this work. While negative opinion spam is more hazardous to a brand's fame compared to positive ones, and thus identifying fake negative reviews might be more crucial, one potential difficulty of our approach is that genuine extremely negative reviews written by renowned reviewers are much more sparse than extremely positive ones, especially for famous products, such as the most popular Chicago hotels in the op_spam_v1.3 dataset.

Acknowledgments

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada and by the University of Toronto. We would like to thank Myle Ott from Cornell University for kindly providing us with the op_spam_v1.3 dataset.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012a. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175, Jeju Island, Korea, July. Association for Computational Linguistics.

Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. 2012b. Distributional footprints of deceptive product reviews. In *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM 2012)*, June.

Stephanie Gokhman, Jeff Hancock, Poornima Prabhu, Myle Ott, and Claire Cardie. 2012. In search of a gold standard in studies of deception. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 23–30, Avignon, France, April. Association for Computational Linguistics.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of First ACM International Conference on Web Search and Data Mining (WSDM-2008)*, pages 219–230.

Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 1549–1552.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, New York, NY, USA. ACM.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 423–430, Sapporo, Japan, July.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 939–948.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, Madison, Wisconsin, USA, July.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*, volume 5 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, May.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin K. Tsou. 2011. Multi-aspect sentiment analysis with topic models. In *ICDM Workshops*, pages 81–88.

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

Myle Ott, Claire Cardie, and Jeffrey T. Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Short Papers*, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

James W. Pennebaker, Cindy K. Chung, Molly Ireland, Amy Gonzales, and Roger J. Booth. 2007. The development and psychometric properties of LIWC2007. Technical report, Austin, TX, LIWC.Net.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Victoria L. Rubin and Tatiana Vashchilko. 2012. Identification of truth and deception in text: Application of vector space model to rhetorical structure theory. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 97–106, Avignon, France, April.

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistcs: Human Language Technologies (ACL-08: HLT)*, pages 308–316, Columbus, Ohio, June. Association for Computational Linguistics.

Guangyu Wu, Derek Greene, Barry Smyth, and Pádraig Cunningham. 2010. Distortion as a validation criterion in the identification of suspicious reviews. In *Proceedings of the First Workshop on Social Media Analytics (SOMA 2010)*, pages 10–13, New York, NY, USA.

Kyung-Hyan Yoo and Ulrike Gretzel. 2009. Comparison of deceptive and truthful travel reviews. In *Information and Communication Technologies in Tourism 2009*, pages 37–47. Springer Vienna.