Detecting Positive and Negative Deceptive Opinions using PU-learning

Donato Hernández Fusilier[a,c,*], Manuel Montes-y-Gómez[b],
Paolo Rosso[c], Rafael Guzmán Cabrera[a]

[a]*División de Ingenierías, Campus Irapuato-Salamanca.Universidad de Guanajuato. Mexico*
[b] *Laboratorio de Tecnologías del Lenguaje. Instituto Nacional de Astrofísica, Óptica y Electrónica. Mexico.*
[c]*NLE Lab., PRHLT Research Center, Universitat Politècnica de València. Spain.*

Abstract

Nowadays a large number of opinion reviews are posted on the Web. Such reviews are a very important source of information for customers and companies. The former rely more than ever on online reviews to make their purchase decisions, and the latter to respond promptly to their clients' expectations. Unfortunately, due to the business that is behind, there is an increasing number of deceptive opinions, that is, fictitious opinions that have been deliberately written to sound authentic, in order to deceive the consumers promoting a low quality product (positive deceptive opinions) or criticizing a potentially good quality one (negative deceptive opinions). In this paper we focus on the detection of both types of deceptive opinions, *positive and negative*. Due to the scarcity of examples of deceptive opinions, we propose to approach the problem of the detection of deceptive opinions employing PU-learning. PU-learning is a semi-supervised technique for building a binary classifier on the basis of positive (i.e., deceptive opinions) and unlabeled examples only. Concretely, we propose a *novel* method that with respect to its original version is much more conservative at the moment of selecting the negative examples (i.e., *not* deceptive opinions) from the unlabeled ones. The obtained results show that the proposed PU-learning method *consistently* outperformed the original PU-learning approach. In particular, re-

[*] Corresponding author

Email addresses: donato@ugto.mx (Donato Hernández Fusilier), mmontes@inaoep.mx (Manuel Montes-y-Gómez), prosso@dsic.upv.es (Paolo Rosso), guzmanc@ugto.mx (Rafael Guzmán Cabrera)

sults show an average improvement of 8.2% and 1.6% over the original approach in the detection of positive and negative deceptive opinions respectively.

Keywords:
Opinion Mining, Opinion Spam, Deceptive Opinions, PU-learning

1. Introduction

The Web is not only the greatest repository of digital information ever invented but also the largest communication platform. This characteristic has motivated businesses of all sizes and kinds, such as television networks, film makers, hotels and restaurants, to use the Web as a critical marketing venue by creating websites and discussion forums for their products and services (Duan et al., 2008). With the increasing availability of such review sites and blogs, consumers rely more than ever on online reviews to make their purchase decisions. A recent survey found that 87% of them have reinforced their decisions to purchase a product or service by positive online reviews. At the same time, 80% of consumers have also changed their minds about purchases based on negative information they found online[1].

Detecting opinion spam is a very challenging problem since opinions expressed on the Web are typically short texts, written by unknown people using different styles and for different purposes. Opinion spam has many forms, e.g., fake reviews, fake comments, fake blogs, fake social network postings and deceptive texts. Opinion spam reviews may be detected by methods that seek for duplicate reviews (Jindal and Liu, 2008); however, this kind of opinion spam only represents a small percentage of the opinions from review sites. In this paper we focus on a potentially more insidious type of opinion spam, namely, *deceptive opinion spam*, which consists of fictitious opinions that have been deliberately written to sound authentic in order to deceive the consumers.

The detection of deceptive opinion spam has been recently solved by means of supervised text classification techniques. These techniques have demonstrated to be very robust if they are trained using large sets of labeled instances from both classes, deceptive and truthful opinions. For example, some works have reported F_1 measures around 0.90 (Ott et al., 2011, 2013; Feng and Hirst, 2013). Nevertheless, in real application scenarios it is very difficult to construct such

[1] How Online Reviews Affect Your Business. http://mwpartners.com/positive-online-reviews. Visited: April 2, 2014.

large training sets and, much more important, it is almost impossible to determine the authenticity of the opinions, i.e., to assemble a set of verified truthful reviews (Mukherjee et al., 2011). In order to meet this restriction in this paper we propose to apply *PU-learning* (Liu et al., 2002) to detect deceptive opinion spam in order to be able to learn only from a few examples of deceptive opinions and a set of unlabeled data, under the consideration that deceptive opinion spam can be accurately generated using a Mechanical Turk crowdsourcing service as suggested by Ott et al. (2011).

The PU-learning approach was originally used and evaluated in thematic text classification, in problems showing high cohesion among the documents from the target (positive) class, and having great diversity in the unlabeled subset (Liu et al., 2002, 2003). The main contribution of this paper is the proposal of a conservative variant of the original method by Liu et al. (2002) that is especially suited to the task of detection of opinion spam, where deceptive opinions are very diverse in content and style, and there are only slightly differences between deceptive and truthful opinions.

The evaluation of the proposed method was carried out using a set of hotel reviews gathered by Ott et al. (2013) containing *positive* and *negative* deceptive opinion spam[2]. The results are encouraging; on the one hand, they indicate that using only a hundred of examples of deceptive opinions for training it is possible to reach classification F_1 measures of 0.8 and 0.7 for positive and negative opinions respectively. On the other hand, they demonstrate the appropriateness of the proposed PU-learning variant for detecting opinion spam, since its results significantly outperformed those from the original approach in both kinds of opinion spam. As a further contribution, in a last experiment we analysed the role of opinions' polarity in the detection of deception. Our results confirm that negative deceptive opinions are more difficult to detect than positive spam, but they also show that having one single classifier for analysing both kinds of opinions is better that using two separate classifiers, suggesting that there are common characteristics in the way people write positive and negative opinion spam.

The rest of the paper is organized as follows. Section 2 introduces some related works in the field of opinion spam detection. Section 3 describes our adaptation of the PU-learning approach to the task of opinion spam detection. Section 4 presents the different opinion spam datasets used in the experiments. Section 5 describes the experimental settings and presents the results from the classification

[2] http://myleott.com/op_spam

of deceptive and truthful reviews in several sets of positive and negative opinions. Finally, Section 6 presents our conclusions and discusses some future work directions.

2. Related Work

The detection of spam on the Web has been mainly approached as a binary classification problem (spam vs. non-spam). It has been traditionally studied in the context of e-mail (Drucker et al., 2002), and Web pages (Gyongyi et al., 2004; Ntoulas et al., 2006). The detection of opinion spam, i.e., the identification of fake reviews that try to deliberately mislead human readers, is just another face of the same problem (Raymond et al., 2011). Nevertheless, the construction of automatic detection methods for this task is more complex than for the others since manually gathering labeled reviews –particularly truthful opinions– is very hard, if not impossible (Mukherjee et al., 2011).

Due to the lack of reliable labeled data, most initial works regarding the detection of opinion spam considered unsupervised approaches which relied on meta-information from reviews and reviewers. For example, Jindal and Liu (2008) proposed detecting opinion spam by identifying duplicate content. Although this method showed good precision in a review data set from Amazon, it has the disadvantage of under detecting original fake reviews. It is well known that spammers modify or paraphrase their own reviews to avoid being detected by automatic tools. In a subsequent paper, Jindal et al. (2010) proposed to detect spammers by searching for unusual review patterns; for example, they classify a reviewer as spam suspect if he wrote negative reviews about all the products of a brand but wrote positive reviews about a competing brand.

In this same category of unsupervised approaches, Mukherjee et al. (2011) proposed a method for detecting groups of opinion spammers based on criteria such as the number of products for which the group work together and a high content similarity of their reviews. Similarly, in (Wu et al., 2010) the authors present a method to detect hotels which are more likely to be involved in spamming. They proposed a number of criteria that might be indicative of suspicious reviews and evaluated alternative methods for integrating these criteria to produce a suspiciousness ranking. Their criteria mainly derive from characteristics of the network of reviewers and also from the impact and ratings of reviews. It is worth mentioning that they did not take advantage of reviews' content for their analysis. Finally, in a recent work by Sihong et al. (2012), it has been demonstrated that a high correlation between the increase in the volume of singleton reviews and a sharp increase or decrease in the ratings is a clear signal that the rating is manipulated by possible spam reviews. Supported by this observation they proposed a spam detection method based on temporal pattern discovery.

It was only after the release of the gold-standard datasets by Ott et al. (2011, 2013), which contain examples of positive and negative deceptive opinion spam, that it was possible to conduct supervised learning and a reliable evaluation of the task. Ott et al. (2011) constructed a SVM classifier to distinguish between *positive* deceptive and truthful reviews using different stylistic, syntactic and lexical features. Then, in (Ott et al., 2013) they applied the same approach to classify *negative* opinions. The main conclusion from these works is that standard text categorization techniques using unigrams and bigrams word features are effective at detecting deception in text, and that their results significantly outperform those from human judges. Following this research direction, Feng et al. (2012a,b) 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. Their experimental results consistently find statistical evidence that deep syntactic patterns are helpful in discriminating deceptive writing. Similarly, Feng and Hirst (2013) extended Ott et al. and Feng et al.'s works by incorporating features that characterize 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. This idea was supported on 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. This approach showed to significantly improve the performance of identifying deceptive reviews.

The method proposed in this paper is similar to the above-mentioned works in the sense that it also applies a supervised approach to automatically identify deceptive and truthful reviews. However, all these methods exhibit a key problem: they depend on the availability of large amounts of labeled examples of deceptive and truthful opinions. This is particularly evident for the last two works which look for syntactic patterns and profile features. In order to overcome this limitation and be able to deal with real application scenarios, in (Hernández et al., 2013) we proposed a method that learns only from a few examples of deceptive opinions and a set of unlabeled data. Specifically, we have evaluated the feasibility of detecting positive deceptive opinions with PU-learning. This paper extends our previous work in four ways: it compares the performance of the proposed approach and the original PU-learning method in the classification of deceptive opinion spam; it reports additional experimental results on a set of negative deceptive opinions, showing the proficiency of the method to deal with opinion spam of both polarities; it studies the role of opinions' polarity in the detection of deception; lastly, it presents an analysis of the performance of the method when using word unigrams and bigrams as features as well as different classifiers, particularly SVM and Naïve Bayes.

3. PU-Learning for Opinion Spam Detection

PU-learning is a semi-supervised technique for building a binary classifier based on positive and unlabeled examples only (Liu et al., 2002, 2003). In PU-learning, two sets of examples are available for training: the set P of positive instances, and a set U, which is assumed to contain a mixture of both positive and negative examples, but without any label. This contrasts with other forms of semi-supervised learning, where it is assumed that the training set contains labeled examples of both classes. In our particular problem, P corresponds to the set of labeled deceptive opinions, and U is a set of unlabeled review opinions –presumably– containing a combination of deceptive and truthful opinions.

The basic algorithm for PU-learning as described in (Liu et al., 2002, 2003) is shown in Algorithm 1. From now on we will refer to this algorithm as *original* PU-learning. The first part of this algorithm (from line 1 to 6) considers the identification of a initial set of reliable negative instances from U. It proceeds as follows: first, the whole unlabeled set U is considered as the negative class, and a classifier is trained using this set in conjunction with the set P of positive examples. Then, this classifier is used to classify (i.e., automatically label) the unlabeled set U. The instances from the unlabeled set classified as negative are selected to form the initial set of reliable negative instances (RN). The second part of the algorithm (from line 7 to 13) iteratively enlarges the set of reliable negative instances by aggregating some additional instances from U. This is done by training a binary classifier using the sets P and RN (from the previous iteration), and classifying the remaining instances at U. The instances from U classified as negative (Q) are aggregated to the set of reliable negative instances from the previous iteration.

The original PU-learning approach has shown very good performance in text classification (Liu et al., 2002, 2003). It has been observed that its effectiveness is very related to the level of cohesion among the positive examples. Accordingly, in tasks showing high similarity among the positive labeled examples, the PU-learning algorithm tends to do a good initial selection of the reliable negative

```
$i \leftarrow 1$
$C_i \leftarrow Generate\_Classifier(P, U)$
$U_i^L \leftarrow C_i(U)$
$Q_i \leftarrow Extract\_Negatives(U_i^L)$
$RN_i \leftarrow Q_i$
$U_i \leftarrow U - Q_i$
while $|Q_i| > \emptyset$ do
    $i \leftarrow i + 1$
    $C_i \leftarrow Generate\_Classifier(P, RN_{i-1})$
    $U_i^L \leftarrow C_i(U_{i-1})$
    $Q_i \leftarrow Extract\_Negatives(U_i^L)$
    $U_i \leftarrow U_{i-1} - Q_i$
    $RN_i \leftarrow RN_{i-1} + Q_i$
$Return(C_i)$
```

Algorithm 1: Original PU-learning algorithm. P and U are the sets of positive and unlabeled examples respectively; C_i is the binary classifier at iteration i; Q_i represents the set of unlabeled examples from U_i classified as negative by C_i, and RN_i is the set of reliable negative examples gathered from iteration 1 to iteration i.

instances and, iteration by iteration, it is able to enlarge this set with more relevant negative examples.

Motivated by this observation, and by the fact that deceptive opinions are very diverse in content and style, we propose a conservative variant of the original PU-learning algorithm. This new algorithm, herein referred as *modified* PU-learning, assumes that the first classifier will be somewhat imprecise and it may select a potentially very noisy initial set of reliable negative instances. Therefore, instead of following an iterative growing strategy for building the RN set, this method considers its iterative pruning. Algorithm 2 describes the modified PU-learning algorithm. The first part of this algorithm (from line 1 to 6) is the same as in the original algorithm. The second part of the algorithm (from line 7 to 12) is significantly different: it iteratively reduces the set of reliable negative instances by eliminating the less confident instances from RN. This is done by training a binary classifier using the sets P and RN (from previous iteration), and classifying the instances at RN. The instances classified as positive are eliminated from it, forming in this way a new small set of reliable negative instances. Line 7 from the algorithm indicates the new stop condition. The purpose of this condition is two-fold: on the one hand, to ensure a continuous but gradual reduction of the instances from the unlabeled set used as negative examples, and, on the other hand, to avoid a high imbalance in the training set by a radical reduction of RN. By means of this condition it is possible to identify a few number of high quality negative instances from the unlabeled set, and to construct a better final binary classifier than using the original PU-learning approach.

4. Datasets

The evaluation of the proposed method was carried out using the corpora assembled by Ott et al. (2011, 2013). These corpora include a total of 1600 labeled examples of deceptive and truthful review opinions about the 20 most popular Chicago hotels[3]. The corpora is organized as follows: 400 truthful positive reviews, 400 truthful negative reviews, 400 deceptive positive reviews and 400 deceptive negative reviews. Deceptive opinions were generated using the Amazon Mechanical Turk, whereas (likely) truthful opinions were mined from reviews on TripAdvisor, Expedia, Hotels.com, Orbitz, Priceline, and Yelp. The following paragraphs show two positive opinions taken from (Ott et al., 2011). These

[3] http://myleott.com/op_spam

```
$i \leftarrow 1;$
$C_i \leftarrow Generate\_Classifier(P, U)$
$U_i^L \leftarrow C_i(U)$
$Q_i \leftarrow Extract\_Negatives(U_i^L)$
$RN_i \leftarrow Q_i$
$Q_0 \leftarrow Q_i$
while $(|Q_i| <= |Q_{i-1}|$ and $|P| < |RN_i|)$ do
    $i \leftarrow i + 1$
    $C_i \leftarrow Generate\_Classifier(P, RN_{i-1})$
    $RN_i^L \leftarrow C_i(RN_{i-1})$
    $Q_i \leftarrow Extract\_Negatives(RN_i^L)$
    $RN_i \leftarrow Q_i$
$Return(C_i)$
```

Algorithm 2: Modified PU-learning algorithm. P and U are the sets of positive and unlabeled examples respectively; Q_i and RN_i represent the sets of identified and retained reliable negative examples at iteration i, and C_i is the binary classifier at iteration i.

examples are very interesting since they show the great complexity of the automatically –and even manually– detection of deceptive opinions. Both opinions are very similar and just minor details can help distinguishing one from the other. For example, in their research Ott et al. (2011) found that there is a relationship between deceptive language and imaginative writing, and that deceptive reviews tend to use the words ”experience”, ”my husband”, ”I”, ”feel”, ”business”, and ”vacation” more than genuine ones.

Example of a positive *deceptive* opinion

> My husband and I stayed for two nights at the Hilton Chicago, and enjoyed every minute of it! The bedrooms are immaculate, and the linens 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.

Example of a positive *truthful* opinion

> 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 concierge 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 Ave exit. It's a great view.

In order to simulate real scenarios to evaluate the performance of the proposed PU-learning method we assembled several different datasets from Ott et al.'s corpora. These datasets contain opinions from both polarities and different number of labeled samples for training. The following paragraphs describe their construction. It is worth mentioning that for the experiments we built *five* different examples for each subset configuration, and that we always report their average results.

Datasets of positive opinions: From the set of 400 deceptive and 400 truthful positive opinions from Ott et al.'s corpora, we first randomly selected 80 deceptive opinions and 80 truthful opinions to build a fixed test set. Then, the remaining 640 opinions were used to build six training sets of different sizes and distributions. They contain 20, 40, 60, 80, 100 and 120 positive instances (deceptive opinions) respectively. In all cases we used a set of 520 unlabeled instances containing a distribution of 320 truthful opinions and 200 positive deceptive opinions.

Datasets of negative opinions: Their construction was similar to the positive datasets but using the set of 400 deceptive and 400 truthful negative opinions from Ott et al.'s corpora. Accordingly, we randomly selected 80 negative deceptive opinions and 80 negative truthful opinions to build the test set. Then, the remaining 640 negative opinions were used to build six training sets of different sizes and distributions. They contain 20, 40, 60, 80, 100 and 120 negative deceptive opinions (positive instances) respectively. In all cases it was used a set of 520 unlabeled instances containing a distribution of 320 negative truthful opinions and 200 negative deceptive opinions.

Datasets of mixed polarity: These datasets were built to analyse the role of polarity in the detection of opinion spam. They were mainly assembled by combining the positive and negative sets previously described. Therefore, we form a test set consisting of 160 deceptive and 160 truthful opinions, and using the remaining 1280 opinions we built six training sets containing 40, 80, 120, 160, 200 deceptive opinions respectively (half of them positive opinions and the other half negative). In all cases it was used a set of 1040 unlabeled instances containing a distribution of 640 truthful opinions and 400 deceptive opinions.

5. Experimental Evaluation

5.1. Experimental settings

Document preprocessing: We removed all punctuation marks and numerical symbols, i.e., we only considered alphabetic tokens. We maintained the stop words, and converted all words to lowercase letter. These operations were applied on both labeled and unlabeled documents.

Learning algorithms: We used the Naïve Bayes (NB) classifier for all the experiments. We employed the implementation by Weka (Hall et al., 2009), considering all words occurring more than once in the training set as features. For the reported experiments we applied a binary weighting scheme. Additionally, in Section 5.5, we report results from a SVM classifier considering word unigrams and bigrams as features as suggested by Ott et al. (2011, 2013). For this experiment we also employed the SVM implementation by Weka using a linear kernel and default parameters.

Evaluation measure: The evaluation of the effectiveness of the proposed method was carried out by means of the macro average of the F_1 measure for both classes, deceptive and truthful opinions. As mentioned before, in all the experiments we report the average results on the five different examples for each subset configuration of the datasets. The F_1 measure for each opinion category O_i is computed as follows:

$$f - measure(O_i) = \frac{2 \times recall(O_i) \times precision(O_i)}{recall(O_i) + precision(O_i)} \tag{1}$$

$$recall(O_i) = \frac{number\ of\ correct\ predictions\ of\ O_i}{number\ of\ opinions\ of\ O_i} \tag{2}$$

$$precision(O_i) = \frac{number\ of\ correct\ predictions\ of\ O_i}{number\ of\ predictions\ as\ O_i} \tag{3}$$

Statistical comparison of methods: Following the recommendation by Demšar, J. (2006), we used the Wilcoxon Signed Ranks Test for comparing our method against other classification approaches. For these comparisons, we considered a 95% level of significance (i.e., $\alpha = 0.05$) and a null hypothesis that both algorithms perform equally well. It is important to mention that for comparing any two methods, we created two distributions with 20 values each, corresponding to their results in 5 folds from 4 collections (60, 80, 100 and 120 training instances).

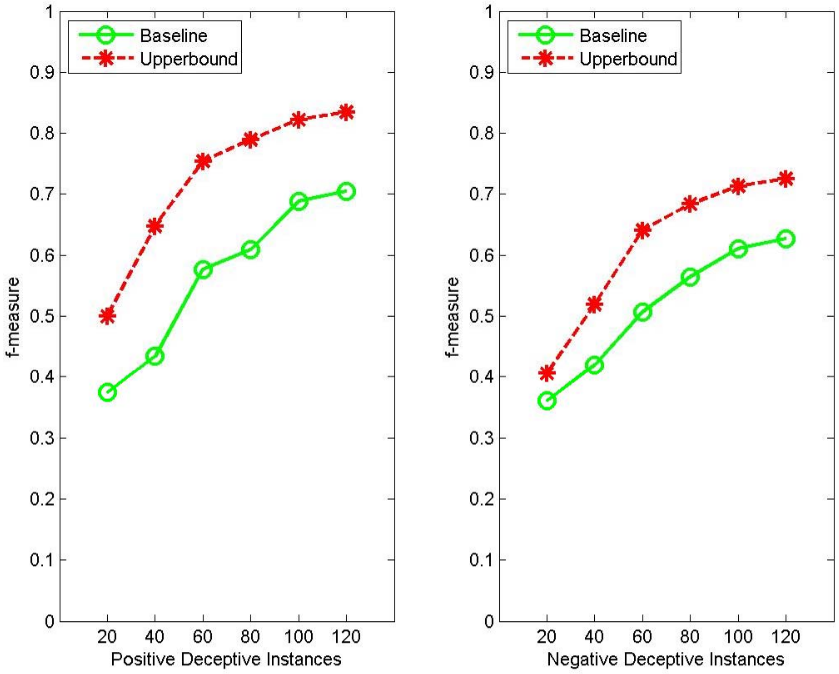

Figure 1: Baseline and Upperbound results for the different subsets of positive and negative opinions.

5.2. Experiment 1: Lower and upper bounds for the PU-learning approach

This first experiment focused on evaluating the detection of positive and negative opinion spam under more realistic conditions, which consider only a few labeled deceptive opinions (and a set of unlabeled data) for training a classifier. The main objective of this experiment was to analyse the feasibility of the PU-learning approach for handling these complex but realistic scenarios.

This analysis was done using the first two datasets described in Section 4. As baseline we considered the results that were obtained by training a NB classifier using the whole unlabeled set as the negative class[4]. This is a simple but common approach to build a binary classifier in case of lack of negative instances. It is worth mentioning that these results correspond to the results from the first iteration of the PU-learning approach. Moreover, as ideal performance of the PU-learning approach we considered the results that were obtained by training the NB classifier using only the truthful instances from the unlabeled set as the negative class. These results represent the upperbound for the proposed method since they

[4] Notice that in all our experiments the set of deceptive opinions are positive, negative or a combination of both, is used as the positive class.

could be reached only if the set of reliable negative instances is perfectly identified from the rest of the unlabeled instances. Figure 1 shows these two kinds of results for the different training subsets of datasets of positive and negative opinions.

Results from Figure 1 clearly indicate that classifying negative opinions is more difficult than the detection of positive deceptive and truthful opinions; the highest F_1 measure obtained for negative opinions was 0.74, whereas for positive opinions the ideal PU-learning approach could obtained a $F_1 = 0.85$. Furthermore, the improvement in the classification performance achieved by the PU-learning approach over the baseline was greater for positive opinions (30%) than for negatives (19%). This tendency confirm previous work's conclusions, which also suggest that negative spam is more complex for being identified.

Another interesting observation from Figure 1 is that PU-learning was incapable to learn a suitable classifier when having very few labeled deceptive opinions for training. Baseline results were lower than 0.5 when using 20 and 40 labeled examples, indicating that the initial selection of the reliable negative instances is very difficult under such circumstances. On the other hand, the upper-bound results were also not good; its poor performance could be attributed to two main reasons: the great imbalance in the training sets (20 or 40 deceptive opinions against 320 truthful opinions), and the difficulty of capturing the diversity in content and style of deceptive opinions from a small number of examples.

5.3. Experiment 2: Original vs modified PU-learning

This experiment focused on the comparison of the original and modified PU-learning methods in the classification of deceptive and truthful opinions. Figure 2 presents a general overview of the results obtained by these two approaches using training sets of positive and negative opinions of different sizes. These results show that the proposed PU-learning method *sistematically* outperformed baseline results as well as the results from the original PU-learning approach. In particular, it shows an average improvement of 8.2% and 1.6% over the original approach in the detection of positive and negative deceptive opinions respectively. Using the Wilcoxon test as explained in Section 5.1, we found that the proposed PU-learning approach is significantly better that both the baseline and original PU-learning method with $p < 0.05$ in both polarities.

Results from Figure 2 corroborate the already reported complexity involved in the classification of negative opinions; for this kind of opinions the best result of the proposed method was $F_1 = 0.7$ using 120 labeled deceptive opinions for training. In contrast, our method achieved a $F_1 = 0.79$ in the detection of positive

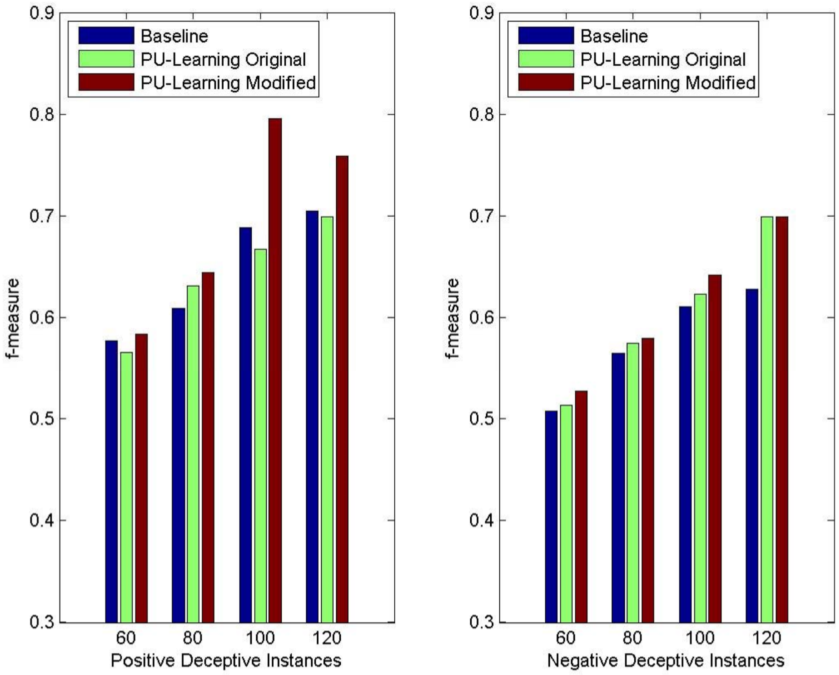

Figure 2: Results of the baseline, original PU-learning, and modified PU-learning in the classification of deceptive and truthful opinions from both polarities.

deceptive and truthful opinions using only 100 labeled training samples. Searching for an explanation for this behaviour, we noticed that the vocabulary employed in negative opinions was larger than the vocabulary from positives, indicating that their content is in general more detailed and diverse, and, therefore, that there larger training sets are needed for their adequate modelling.

Additional detailed results from this experiment are shown in Tables 1 and 2. These tables include the precision, recall and f-measure of the classification of deceptive as well as truthful opinions. They also show information about the number of iterations done by both PU-learning algorithms as well as the distribution of the training sets built by each of them.

In view that our main objective is the detection opinion spam, it is of particular interest to analyse the classification results corresponding to the positive class (i.e., deceptive opinions). Table 1 shows a very good performance in the detection of positive deceptive opinions; whereas the original PU-learning approach obtained a maximum result of $F_1 = 0.626$, the proposed PU-learning method reached a $F_1 = 0.78$, giving an improvement of 24.6%. Furthermore, this result presents a good trade-off between precision (0.85) and recall (0.72), compromise

Table 1: Detailed results on the classification of *positive* opinions using 60, 80, 100 and 120 labeled deceptive opinions (DP) and 520 of unlabeled examples (UN) for training. In this table, P, R and F state for precision, recall and f-measure respectively.

Initial Training Set	**Used Method**	**Deceptive**			**Truthful**			**General F-measure**	**# of iterations**	**Final Training Set**
		P	**R**	**F**	**P**	**R**	**F**			
60-DP/520-UN	BASELINE	0.896	0.268	0.408	0.605	0.975	0.746	0.577	1	60-DP/520-UN
	PU-L ORIGINAL	0.878	0.275	0.413	0.572	0.965	0.718	0.566	2	60-DP/473-UN
	PU-L MODIFIED	0.895	0.298	0.441	0.581	0.968	0.726	0.584	4	60-DP/394-UN
80-DP/520-UN	BASELINE	0.921	0.330	0.482	0.593	0.973	0.736	0.609	1	80-DP/520-UN
	PU-L ORIGINAL	0.925	0.363	0.519	0.604	0.970	0.744	0.632	2	80-DP/450-UN
	PU-L MODIFIED	0.842	0.415	0.547	0.618	0.933	0.742	0.645	7	80-DP/253-UN
100-DP/520-UN	BASELINE	0.919	0.408	0.561	0.621	0.965	0.756	0.689	1	100-DP/520-UN
	PU-L ORIGINAL	0.926	0.420	0.575	0.627	0.968	0.760	0.668	2	100-DP/432-UN
	PU-L MODIFIED	**0.852**	**0.728**	**0.780**	**0.768**	**0.868**	**0.811**	**0.796**	8	100-DP/112-UN
120-DP/520-UN	BASELINE	0.931	0.453	0.606	0.640	0.968	0.770	0.705	1	120-DP/520-UN
	PU-L ORIGINAL	0.916	0.480	0.626	0.648	0.955	0.772	0.699	2	120-DP/425-UN
	PU-L MODIFIED	0.803	0.700	0.743	0.738	0.823	0.774	0.759	7	120-DP/144-UN

that could not be achieved by any of the other methods. On the other hand, as indicated in Table 2, the detection of negative deceptive opinions was not as good as in the case of positive opinions. The best result by the proposed method was $F_1 = 0.657$. However, the average improvement of the proposed method over the original PU-learning approach was of 11% for all training conditions, indicating that the proposed approach is considerably better than the original one in the identification of opinion spam. It is worth mentioning that the better results by the proposed method in both polarities could be explained by its better selection of reliable negative instances. While the original approach retained more than 400 out of 500 instances in the negative class, our approach carried out a very hard selection of instances (i.e., truthful opinions), extracting in some cases less than 200 examples from the unlabeled set. Furthermore, the larger the set of labeled training instances, the higher the reduction made by the proposed method on the set of reliable negative instances. This is in contrast to the original PU-learning approach where the selection of reliable negative instances was uncorrelated with the number of labeled training instances.

5.4. Experiment 3: Polarity and deception under PU-learning

The purpose of this experiment was to analyse the role of polarity in the classification of deceptive and truthful opinions, in the context of the proposed PU-learning method. To carry out this analysis we used the dataset of mixed polarity described in Section 4, and we evaluated the performance of two different classifier configurations. The first configuration considered *one* single classifier for

Table 2: Detailed results on the classification of *negative* opinions using 60, 80, 100 and 120 labeled deceptive opinions (DP) and 520 of unlabeled examples (UN) for training. In this table, P, R and F state for precision, recall and f-measure respectively.

Training Set	**Method Used**	**Deceptive**			**Truthful**			**F-measure general**	**# of iterations**	**Final training set**
		P	**R**	**F**	**P**	**R**	**F**			
60-DN/520-UN	BASELINE	0.906	0.178	0.312	0.548	0.980	0.703	0.508	1	60-DN/520-UN
	PU-L ORIGINAL	0.926	0.195	0.321	0.550	0.985	0.706	0.514	2	60-DN/483-UN
	PU-L MODIFIED	0.932	0.213	0.344	0.556	0.985	0.711	0.528	4	60-DN/404-UN
80-DN/520-UN	BASELINE	0.903	0.270	0.412	0.570	0.968	0.717	0.565	1	80-DN/520-UN
	PU-L ORIGINAL	0.904	0.285	0.429	0.575	0.965	0.720	0.575	2	80-DN/464-UN
	PU-L MODIFIED	0.845	0.333	0.446	0.590	0.923	0.713	0.580	5	80-DN/295-UN
100-DN/520-UN	BASELINE	0.926	0.333	0.486	0.593	0.970	0.736	0.611	1	100-DN/520-UN
	PU-L ORIGINAL	0.902	0.360	0.510	0.599	0.955	0.736	0.623	2	100-DN/450-UN
	PU-L MODIFIED	0.825	0.468	0.578	0.612	0.860	0.706	0.642	6	100-DN/202-UN
120-DN/520-UN	BASELINE	0.898	0.370	0.517	0.604	0.953	0.738	0.628	1	120-DN/520-UN
	PU-L ORIGINAL	0.890	0.395	0.543	0.611	0.948	0.740	0.699	2	120-DN/438-UN
	PU-L MODIFIED	**0.788**	**0.595**	**0.657**	**0.672**	**0.803**	**0.723**	**0.699**	6	120-DN/177-UN

detecting both positive and negative opinion spam. In other words, it did not take into account the polarity of reviews. In contrast, the second classifier configuration approached the identification of positive and negative spam as two different problems; it is mainly an *ensemble* of the two independent classifiers evaluated in the previous section. It is important to clarify that the first classifier used all available training data, whereas, in the ensemble configuration, each one of the classifiers was trained using only half of the data.

Table 3: Detecting Deceptive opinions when using 120, 160, 200 and 240 samples of Deceptive opinions and 1040 opinions of mixed polarities in the Unlabeled set (520 Deceptive and 520 Truthful).

Training Set	**Classifier Configuration**	**F-measure**		
		Deceptive Op	**Truthful Op**	**General**
120-D 1040-U	ONE SINGLE CLASSIFIER	0.665	0.714	0.690
	ENSEMBLE TWO CLASSIFIERS	0.392	0.719	0.556
160-D 1040-U	ONE SINGLE CLASSIFIER	0.740	0.797	0.769
	ENSEMBLE TWO CLASSIFIERS	0.496	0.727	0.612
200-D 1040-U	ONE SINGLE CLASSIFIER	0.717	0.761	0.739
	ENSEMBLE TWO CLASSIFIERS	0.679	0.758	0.719
240-D 1040-U	ONE SINGLE CLASSIFIER	0.771	0.790	0.781
	ENSEMBLE TWO CLASSIFIERS	0.700	0.748	0.724

Table 3 shows the results of this experiment. They indicate that for all the cases the configuration based on one classifier outperformed the results from the ensemble configuration; according to the Wilcoxon test, the one single classifier is statistically better than the ensemble in general F_1 measure with $p < 0.05$. It is worth noting that the advantage shown by the single classifier was particularly relevant for the cases using less training samples (120 and 160 mixed labeled deceptive opinions), in which the improvement was around 25%. These results are quite interesting and unexpected; they show that, despite their clear differences, positive and negative opinions have common elements that a classifier can exploit to enhance the spam modelling and classification. Moreover, the results indicate that in situations with lack of data, as the ones considered in this study, more data, even from a different polarity, it is always useful.

5.5. Experiment 4: On the choice of features and classifier

The goal of this last experiment was to evaluate the variation in the performance of the proposed PU-learning method when using other base classifier and a different set of features. Particularly, we employed a SVM classifier and combination of word unigrams and bigrams as features, such as considered by some previous successful works (Ott et al., 2011, 2013).

Table 4 shows the results of this experiment. According to the Wilcoxon Signed Ranks Test, these results indicate that the PU-learning method using NB as base classifier is significativelly better than its variant using the SVM classifier with $p < 0.05$, whatever the set of features was used. Somehow this conclusion was not completely unexpected; Forman and Cohen (2004) presented empirical evidence showing that Naïve Bayes models are often relatively insensitive to a shift in training distribution, and surpass SVM when there is a shortage of positives or negatives.

Regarding the used features results are not equally clear, the combination of unigrams and bigrams obtained better results than unigrams when using the NB classifier, but unigrams were the best features for the SVM classifier. For both configurations the differences in F_1 measure were statistically significant with $p < 0.05$. Although conclusions were slightly different for the two selected classifiers, it is important to point out that the proposed PU-learning method showed improvements to baseline results for the two polarities using any of the classifiers.

6. Conclusions and Future Work

Three are the contributions of this paper: (i) We approached the problem of the detection of deceptive opinions using the PU-learning technique because of

Table 4: Results of the classification of positive and negative opinion spam by Naïve Bayes (NB) and SVM using unigrams and bigramas as features. The values correspond to the F_1 measure for both classes, deceptive and truthful opinions.

Training	**Corpus**	**Positive Opinions**		**Negative opinions**	
Set	**Used**	**Unigrams**	**Uni+Bigrams**	**Unigrams**	**Uni+Bigrams**
60-DN/520-UN	Baseline NB	0.577	0.604	0.508	0.579
	PU-L + NB	0.584	0.615	0.528	0.628
	Baseline SVM	0.419	0.344	0.420	0.341
	PU-L + SVM	0.443	0.360	0.433	0.344
80-DN/520-UN	Baseline NB	0.609	0.669	0.565	0.619
	PU-L + NB	0.645	0..686	0.580	0.649
	Baseline SVM	0.472	0.367	0.464	0.358
	PU-L + SVM	0.479	0.367	0.474	0.355
100-DN/520-UN	Baseline NB	0.689	0.691	0.611	0.650
	PU-L + NB	0.796	0.712	0.642	0.700
	Baseline SVM	0.502	0.406	0.503	0.379
	PU-L + SVM	0.539	0.410	0.524	0.387
120-DN/520-UN	Baseline NB	0.705	0.730	0.628	0.680
	PU-L + NB	0.759	0.778	0.699	0.727
	Baseline SVM	0.558	0.442	0.531	0.417
	PU-L + SVM	0.579	0.442	0.616	0.645

the scarcity of deceptive examples we believe it is the most adequate way; (ii) We proposed a novel, more conservative at the time of selecting the reliable negative examples, PU-learning approach; (iii) We analysed the role of the opinions' polarity in the detection of deception. The evaluation of the proposed method was carried out using the standard-de-facto hotel reviews dataset described in Ott et al. (2013) that contains both *positive* and *negative* deceptive opinions. The results are encouraging and indicate that using only a hundred of examples of deceptive opinions for training it is possible to reach F_1 measures of 0.8 and 0.7 for positive and negative deceptive opinions respectively. They show the appropriateness of the proposed PU-learning conservative variant for detecting opinion spam, since its results consistently outperformed those obtained with the original approach in both kinds of deceptive opinions. In a further experiment where the role of opinions' polarity in the detection of deception is analysed, the obtained results confirm that negative deceptive opinions are more difficult to detect than positive ones, but they also show that having one single classifier for analysing both types of deceptive opinions is better that using two separate classifiers, suggesting that there are common characteristics in the way people write positive and negative deceptive opinions.

As future work we aim at applying the novel PU-learning for detecting deceptive language to approach problems such as the detection of online sexual predators as well as the detection of lies in general.

Acknowledgments

This work is the result of the collaboration in the framework of the WIQEI IRSES project (Grant No. 269180) within the FP 7 Marie Curie. The work of the third author was in the framework the DIANA-APPLICATIONS-Finding Hidden Knowledge in Texts: Applications (TIN2012-38603-C02-01) project, and the VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems.

References

Demšar, J.(2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.*, December 2006, Volume:7, Page(s) 1-30.

Drucker, H., Wu, D., & Vapnik, V. N. (2002).Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, August 2002, Volume:10, Issue: 5, Page(s) 1048-1054.

Duan, W., Gu, B., & Whinston, A. (2008). Do Online Reviews Matter? An Empirical Investigation of Panel Data. *Decision Support Systems*, November 2008, Volume: 45, Issue: 4, Page(s) 1007-1016 .

Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic Stylometry for Deception Detection. *ACL '12, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, July 2012, Volume: 2, Page(s) 171-175.

Feng, S., Xing, L., Gogar, A., & Choi, Y. (2012). Distributional Footprints of Deceptive Product Reviews. *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, June 2012, Page(s) 98-105.

Feng, V. W., & Hirst, G. (2013). Detecting deceptive opinions with profile compatibility. *Proceedings of the 6th International Joint Conference on Natural Language Processing*, October 2013, Page(s) 338-346.

Forman, G. and Cohen, I. (2004). Learning from Little: Comparison of Classifiers Given Little Training. *Knowledge Discovery in Databases: PKDD 2004, Lecture Notes in Computer Science*,September 2004, Volume: 3202, Page(s) 161-172.

Gyongyi, Z., Garcia-Molina, H., & Pedersen, J. (2004). Combating Web Spam with Trust Rank. *VLDB '04, Proceedings of the Thirtieth international conference on Very large data bases*, September 2004, Volume: 30, Page(s) 576-587.

Hall, M., Eibe, F., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA Data Mining Software: an Update. *ACM SIGKDD Explorations Newsletter*, June 2009, Volume: 11, Issue: 1, Page(s) 10-18.

Hernández-Fusilier, D., Guzmán-Cabrera, R., Montes-y-Gómez, M., & Rosso, P. (2013). Using PU-learning to Detect Deceptive Opinion Spam. *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, June 2013, Page(s) 38-45.

Jindal, N., & Liu, B. (2008). Opinion Spam and Analysis. *WSDM '08, Proceedings of the 2008 International Conference on Web Search and Data Mining*, February 2008, Page(s) 219-230.

Jindal, N., Liu, B., & Lim, E. (2010). Finding Unusual Review Patterns Using Unexpected Rules. *CIKM '10, Proceedings of the 19th ACM international conference on Information and knowledge management*, October 2010, Page(s) 219-230.

Liu, B., Dai, Y., Li, X. L., Lee, W. S., & Philip, Y. (2002). Partially Supervised Classification of Text Documents. *ICML 2002, Proceedings of the Nineteenth International Conference on Machine Learning*, July 2002, Page(s) 387-394.

Liu, B., Dai, Y., Li, X. L., Lee, W. S., & Philip, Y. (2003). Building Text Classifiers Using Positive and Unlabeled Examples. *ICDM 2003, Third IEEE International Conference on Data Mining*, November 2003, Page(s) 179-186.

Mukherjee, A., Liu, B., Wang, J., Glance, N. & Jindal, N. (2011). Detecting Group Review Spam. *WWW '11, Proceedings of the 20th international conference companion on World wide web*, March 2011, Page(s) 93-94.

Ntoulas, A., Najork, M., Manasse, M., & Fetterly, D. (2006). Detecting Spam Web Pages through Content Analysis. *WWW '06, Proceedings of the 15th international conference on World Wide Web*, May 2006, Page(s) 83-92.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding Deceptive Opinion Spam by any Stretch of the Imagination. *HLT '11, Proceedings of the 49th*

Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, June 2011, Volume: 1, Page(s) 309-319.

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative Deceptive Opinion Spam. *NAACL-HLT 2013, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2013, Page(s) 497-501.

Raymond, Y. K., Lau, S. Y., Liao, R., Chi-Wai, K., Kaiquan, X., Yunqing, X., & Yuefeng, L. (2011). Text Mining and Probabilistic Modeling for Online Review Spam Detection. *ACM Transactions on Management Information Systems*, December 2011, Volume: 2, Issue: 4, Article: 25, Page(s) 1-30.

Sihong, X., Guang, W., Shuyang, L., & Philip, S. Y. (2012). Review Spam Detection via Temporal Pattern Discovery. *KDD '12, Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, August 2012, Page(s) 823-831.

Wilcoxon F.(1945). Individual comparisons by ranking methods. *Biometrics Bulletin, International Biometric Society, Vol. 1 No. 6.*, December 1945, 80-83.

Wu, G., Greene, D. & Cunningham, P. (2010). Merging Multiple Criteria to Identify Suspicious Reviews. *RecSys '10, Proceedings of the fourth ACM conference on Recommender systems*, September 2010, Page(s) 241-244.