information

MDPI

Article

CoSpa: A Co-training Approach for Spam Review Identification with Support Vector Machine

Wen Zhang [1,2,*], Chaoqi Bu [1], Taketoshi Yoshida [3] and Siguang Zhang [4]

1 Research Center on Big Data Sciences, Beijing University of Chemical Technology, Beijing 100029, China; buchaoqi@163.com
2 School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China
3 School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Ashahidai, Nomi City, Ishikawa 923-1292, Japan; yoshida@jaist.ac.jp
4 Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, China; zhangsiguang@casipm.ac.cn
* Correspondence: zhangwen@mail.buct.edu.cn; Tel./Fax: +86-10-6419-7040

Academic Editor: Willy Susilo
Received: 29 January 2016; Accepted: 26 February 2016; Published: 9 March 2016

Abstract: Spam reviews are increasingly appearing on the Internet to promote sales or defame competitors by misleading consumers with deceptive opinions. This paper proposes a co-training approach called CoSpa (Co-training for Spam review identification) to identify spam reviews by two views: one is the lexical terms derived from the textual content of the reviews and the other is the PCFG (Probabilistic Context-Free Grammars) rules derived from a deep syntax analysis of the reviews. Using SVM (Support Vector Machine) as the base classifier, we develop two strategies, CoSpa-C and CoSpa-U, embedded within the CoSpa approach. The CoSpa-C strategy selects unlabeled reviews classified with the largest confidence to augment the training dataset to retrain the classifier. The CoSpa-U strategy randomly selects unlabeled reviews with a uniform distribution of confidence. Experiments on the spam dataset and the deception dataset demonstrate that both the proposed CoSpa algorithms outperform the traditional SVM with lexical terms and PCFG rules in spam review identification. Moreover, the CoSpa-U strategy outperforms the CoSpa-C strategy when we use the absolute value of decision function of SVM as the confidence.

Keywords: co-training; PCFG; spam review; CoSpa; support vector machine

1. Introduction

With the spread of the Internet and Web 2.0, it has been accepted that user-generated content (UGC) contains valuable clues that can be exploited for various applications such as e-commerce [1], tourism [2], software development [3], *etc.* Online reviews, which refer to UGCs of opinions of products that users have purchased via e-commerce, are very popular nowadays. These reviews are widely used by potential customers to seek opinions of existing users before making a purchase decision. For example, if one has to choose between two products of the same type, he or she is very likely to buy the product with more positive reviews from existing users. Therefore, positive reviews can boost the reputation of a product and thus create a competitive advantage over other products. Negative reviews can damage the reputation of a product and thus results in a competitive loss.

For this reason, some companies strive to advocate their products with positive reviews and defame competitors' products with negative reviews. Meanwhile, due to a lack of strict scrutiny, anyone can write either negative or positive reviews on the Web with little payment but possible great monetary gain. In the age of Web 2.0, this could cause a large amount of low quality and even deceptive reviews of some products. In this paper, we use the terms "spam review" and "deceptive review" interchangeably and define a spam review as those online reviews submitted by opinion spammers [4]

to mislead customers for monetary purposes. Spam review here does not include advertisements and meaningless fragments. In an e-commerce environment, most customers can rely merely on users' reviews to decide whether or not the product is worthwhile. Thus, there are an increasing number of customers who are worrying about fake or biased product reviews. This explains why spam review identification has become an attractive problem in the research field.

To automatically identify spam reviews, Ott *et al.* [5] used LIWC2007 (Linguistic Inquiry and Word Count) software [6] to produce psychological features from reviews and combine these features with n-grams to automatically categorize spam opinions using SVM as the machine learning classifier. Their experiments on the spam dataset [5] reported that a combination of psychological features and n-grams can produce accuracy as high as 90% in spam opinion categorization and is much better than human judges of accuracy at 60%. They observed that truthful opinions tend to include more censorial and concrete language than deceptive opinions and are more specific about spatial configurations. Feng *et al.* [7] used not only part-of-speech (POS) features but also Context-Free Grammar (CFG) features to improve spam review identification. Their experiments on four datasets—the spam dataset, Yelp dataset, Heuristic TripAdvisor dataset, and Essays dataset—showed that the deep syntax feature can significantly improve spam opinion detection in comparison with the baseline method of combination of words and POS (part-of-speech) features. Similar work in the field of automatic spam review identification includes Feng and Hirst [8], Zhou *et al.* [9], and Li *et al.* [10].

To train a model to identify spam reviews automatically, we need "enough" labeled reviews to train a learning model to predict the labels of unlabeled reviews. However, in practice, the labeled reviews are expensive to obtain because the task of labeling reviews is labor intensive and time-consuming. This difficulty prompted Jindal and Liu [11] to simply use duplicate reviews as spam reviews and Li *et al.* [12] to use heuristics to establish the golden set in automatic spam review identification. However, it is well recognized that the unlabeled reviews data are more widely available on the Internet and easier to obtain than labeled ones. For instance, on the Amazon website, there are more than 10,000 reviews posted just in the electronics category each month [13] and there are about 225 million reviews in TripAdvisor [14]. For most of these reviews, we do not know exactly whether a given product review is spam or not.

Inspired by this observation, we propose to use a co-training approach [15] to make use of unlabeled data to improve the performance of automatic spam review identification in this paper. We use two types of representations for each review: one is the lexical terms derived from textual contents and the other one is the PCFG rules derived from deep syntax analysis of the reviews. We note that the second view of the CoSpa algorithm is clearly independent of the first view, as the PCFG rules are dependent on lexical terms to some extent. However, PCFG rules are proven to be an informative source for identifying deceptive reviews in Section 3.1 because the sentences used in deceptive reviews are somehow imaginative and different from those used in truthful reviews [16,17]. Previous studies [18,19] showed that if the independence assumption of the co-training approach is violated, the co-training approach can yield negative results. However, our results show that co-training can produce improvement in deceptive review identification even though the two views are not independent. This is also validated by other researchers such as Wang *et al.* [20] in identifying event description and Zhou *et al.* [21] in image retrieval. In fact, it is very difficult to obtain sufficient but redundant views in real practice and to prove that two views are completely independent of each other [22].

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 presents our observation on the spam dataset and proposes the CoSpa algorithm with the CoSpa-C and CoSp-U strategies. Section 4 conducts experiments on the spam dataset and the deception dataset. Section 5 concludes the paper.

2. Related Work

2.1. Co-Training with Multiple Views

Co-training is one of the semi-supervised techniques, first proposed by Blum and Mitchell [15], and has recently been extended into three categories: co-training with multiple views, co-training with multiple classifiers, and co-training with multiple manifolds [23]. In this paper, we only use co-training

with multiple (*i.e.*, two) views for spam review identification. The formal framework of co-training with multiple views can be defined as follows. Suppose we can use two feature sets X_1 and X_2 to describe the same data point x. That is to say, two different "views" on x results in two vectors x_1 and x_2. Further, we assume that the two views X_1 and X_2 are independent of each other and each view in itself can provide enough information for classifying x. That is to say, if we use f_1 and f_2 to denote the classifiers derived from the two views, then we may obtain $f_1(x_1) = f_2(x_2) = l$, where l is the true label of x. In other words, for a data point $x = (x_1, x_2)$, if we derive two classifiers as f_1 and f_2 from the training data, then $f_1(x_1)$ and $f_2(x_2)$ will consistently predict the same label.

2.2. Probabilistic Context-Free Grammar (PCFG)

A context-free grammar (CFG) can be described using a 4-tuple $G = (N, \Sigma, R, S)$, *i.e.*, a finite set N containing non-terminal symbols, a finite set Σ containing terminal symbols, a finite set R containing rules of the form $X \rightarrow Z_1 Z_2 ... Z_n$, and $X \in N$, $n \geqslant 0$, and $Z_i \in (N \cup \Sigma)$ for $i = 1, ..., n$, $S \in N$ is a distinguished start symbol [24]. For instance, the parse tree of the sentence "The rooms were spacious and very fresh" is shown in Figure 1 using the context-free grammar provided by Stanford Parser [25]. Here, N = {S, NP, VP, DT, NNS, VBD, ADJP, JJ, CC, RB}, Σ = {the, rooms, were, spacious, and, very, fresh}, S = S and R= {S→NP VP; NP→DT NNS; VP→VBD ADJP; ADJP→ADJP CC ADJP; ADJP→JJ; ADJP→RB JJ; DT→The; NNS→rooms; VBD→were; JJ→spacious; CC→and; RB→very; JJ→fresh}.

A PCFG consists of a context-free grammar $G = (N, \Sigma, R, S)$ and, for each rule $\alpha \rightarrow \beta \in R$, we have a conditional probability $q(\alpha \rightarrow \beta)$ of choosing rule $\alpha \rightarrow \beta$ in a left-most derivation, given that the non-terminal being expanded is α. τ_G is the set of all possible left-most derivations (parse trees) under the grammar G. For any derivation $t \in \tau_G$, $yield(t)$ denotes the sentence s that is the yield of t. We write $\tau_G(s)$ to refer to the set $\{t : t \in \tau_G, yield(t) = s\}$. That is, $\tau_G(s)$ is the set of all possible parse trees for s. The basic idea in probabilistic context-free grammars is to find a ranking over all possible parsed trees for sentence s in order of probability. That is, to find $t_{\max} = \text{argmax}_{t \in \tau_G(s)} p(t)$ to yield sentence s. Usually, the CKY algorithm or the inside algorithm based on dynamic programming [24] is used to solve the maximization problem.

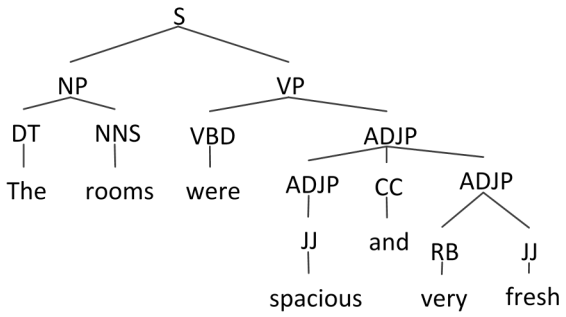


Figure 1. The parse tree of the sentence "The rooms were spacious and very fresh".

In this paper, we follow Feng *et al.* [7] to use four different encodings of production rules based on the PCFG parse trees as follows.

- Rule type 1: unlexicalized production rules (*i.e.*, all production rules except for those with terminal nodes). For instance, the rules of type 1 derived from Figure 1 are [S-NP VP; NP-DT NNS; VP-VBD ADJP; ADJP-ADJP CC ADJP; ADJP-JJ; ADJP-RB JJ].
- Rule type 2: lexicalized production rules (*i.e.*, all production rules). For instance, the rules of type 2 derived from Figure 1 are [DT-The; NNS-rooms; VBD-were; JJ-spacious; CC-and; RB-very; JJ-fresh].
- Rule type 3: unlexicalized production rules combined with the grandparent node. For instance, the rules of type 3 derived from Figure 1 are [ROOT S-NP VP; S NP-DT NNS; S VP-VBD ADJP; VP ADJP-ADJP CC ADJP; ADJP ADJP-JJ; ADJP ADJP-RB JJ].
- Rule type 4: lexicalized production rules (*i.e.*, all production rules) combined with the grandparent node. For instance, the rules of type 4 derived from Figure 1 are [NP DT-The; NP NNS-rooms; VP VBD-were; ADJP JJ-spacious; ADJP CC-and; ADJP RB-very; ADJP JJ-fresh].

2.3. Support Vector Machine (SVM)

As pointed out by Mihalcea [21] and Wan [26], co-training with SVM can produce promising performance in word sense disambiguation and sentiment analysis. In this paper, we also investigate whether it is effective for deceptive review deification. SVM is proposed by Vapnik *et al.* [27] as a promising classification technique. It minimizes the structural risk rather than the empirical risk in classification and is shown to provide higher performance than traditional learning classifiers. Usually, SVM first maps the input data samples into a Hilbert space with higher dimensions than that of the original one using kernel methods [28]. Then, it attempts to find a separating hyperplane that maximizes the margin between positive and negative classes in the high-dimensional space by solving a convex quadratic programming (QP) problem. The solution of the optimal hyperplane can be expressed as a combination of a few data points in the input space called support vectors. Typically, SVM as an optimal hyperplane problem can be written as follows:

$$\min_{\omega,\xi i} \frac{1}{2} ||\omega||^2 + C \sum_i \xi_i, \tag{1}$$

subject to

$$y_i(x_i \times w + b) \geqslant 1 - \xi_i \quad \xi_i \geqslant 0, \forall i, \tag{2}$$

Here, (x_i, y_i) $(1 \leqslant i \leqslant l)$ are the l labeled data points, ω is the slope vector of the hyperplane and ξ_i is the slack variable for the data point (x_i, y_i). After the optimal hyperplane problem is solved, we used the decision function $f(x) = \text{sgn}((\omega, x) + b)$ to classify unlabeled data points. With the intuition that the larger is the distance of a data point from the hyperplane, we are more confident in classifying the data point with its label. We simply used the distance of a data point (x_i, y_i) to the hyperplane $(\omega, x) + b$, *i.e.*, $\frac{|(\omega, x) + b|}{||\omega||}$, as the confidence of (x_i, y_i) to be classified by SVM.

3. CoSpa—The Proposed Approach

3.1. Observation

Using the spam dataset provided by Ott *et al.* [5], we find there is actually a difference in the expressions and PCFG rules used by deceptive and truthful reviews. Table 1 shows the top 10 distinct words used by deceptive reviews and non-deceptive reviews with positive and negative sentiment polarity.

We can see from Table 1 that the words occurring in deceptive reviews are almost completely different from those occurring in truthful reviews. For instance, the words in negative deceptive reviews are almost all general words used to describe a hotel living experience such as "smell", "wife", "money", *etc.*, while for the negative truthful review, the words are more specific and include spatial information such as "floor", "wall", "walk", *etc.* However, we do not see obvious difference between words in negative hotel reviews and positive hotel reviews. We also use a Wilcoxon signed-rank test [29] to investigate the statistical difference between the top 100 words occurring in both deceptive and truthful reviews. Table 2 shows the hypothesis testing results in comparing the words in different paired sets of reviews.

We manually checked the occurrence of the top 100 PCFG rules occurring in the hotel reviews and listed the top 10 distinct PCFG rules in Table 3. Those top 10 distinct rules were found by looking for the different rules with most occurrences in deceptive and ruthful reviews.

Table 1. The top 10 distinct words used by deceptive and truthful hotel reviews. N.D.: term occurring in negative deceptive reviews; N.T.: term is occurring in negative true reviews; P.D.: term is occurring in positive deceptive reviews; P.T.: term is occurring in positive true reviews.

Rank	N.D.	N.T.	P.D.	P.T.
1	Smell (116)	Floor (136)	Visit (121)	Bathroom (116)
2	Luxury (86)	Charge (121)	Experience (104)	Breakfast (106)
3	Recommend (76)	Small (116)	Luxury (104)	Floor (95)
4	Suit (71)	Water (79)	Amazing (89)	Park (89)
5	Wife (69)	Wall (73)	Husband (83)	Bar (77)
6	Money (68)	Phone (73)	Food (81)	Lobby (73)
7	Towel (67)	Bar (72)	Travel (72)	Small (64)
8	Clerk (66)	Shower (68)	Wife (54)	Concierge (60)
9	Weekend (65)	Breakfast (65)	Fantastic (49)	Street (51)
10	Smoke (60)	Walk (59)	Vacation (45)	Coffee (46)

Table 2. The Wilcoxon signed-rank in comparing the paired set of top 100 words in each category of hotel reviews.

p-Value	N.D.	N.T.	P.D.	P.T.
N.D.	-	0.05**	0.23	0.05**
N.T.	0.05**	-	0.05**	0.25
P.D.	0.23	0.23	-	0.05**

**: strong evidence of significance of word difference.

Table 3. The top 10 distinct PCFG rules adopted in deceptive and non-deceptive hotel reviews. N.D.: rule occurring in negative deceptive reviews; N.T.: rule occurring in negative true reviews; P.D.: rule occurring in positive deceptive reviews; P.T.: rule occurring in positive true reviews.

Rank	N. D.	N. T.	P. D.	P. T.
1	VP VBD-was (2000)	ROOT S-NP VP (1848)	TO-to (1268)	S S-NP VP (1219)
2	S S-NP VP (1870)	S-NP VP (4004)	S-VP (1260)	NP PRP-we (886)
3	PRP$-my (737)	VBZ-is (67)	NNP-Chicago (565)	NP-NNS (500)
4	NP PRP$-my (736)	VP VBZ-is (631)	NP PRP$-my (472)	ADJP RB-very (371)
5	VBD-had (628)	NP-NNP (596)	VP SBAR-IN S (390)	NP-NP (367)
6	VP VBD-had (626)	NP CC-and (553)	VP PP-TO NP (361)	JJ-great (353)
7	S VP-MD VP (517)	ADJP-RB JJ (531)	PP NP-NN (356)	NP NP-DT JJ NN (343)
8	VP-VP CC VP (510)	PP NP-NN (526)	NP-NP CC NP (313)	PP IN-on (336)
9	VP PP-TO NP (509)	NP-DT NN NN (509)	VP-MD VP (310)	NP NP-NN (344)
10	WHADVP-WRB (491)	NP-DT NNS (493)	VP-VB NP(303)	VP PP-TO NP (335)

We can see from Table 3 that the PCFG rules with past tense appeared frequently in deceptive reviews such as "VP VBD-was" and "VP VBD-had" to express the past experience of reviewers. For instance, the PCFG rule "VP VBD-was" appeared in "The bed [**was the most uncomfortable thing**] I have ever tried to sleep on". In contrast, truthful reviews more often than not have PCFG rules with present tense to describe some facts such as "VP VBZ-is" in the sentence "Thin walls, and you could hear exactly what the next room [**is talking about**]". Moreover, we also found that in deceptive reviews, "VP (verb phrase)" occurred more frequently than "NP (noun phrase)" but in truthful reviews, the opposite was true. For instance, the PCFG rule "VP-VP CC VP" frequently occurred in deceptive reviews such as "only [to arrive and have] a room without a view . . . ." The PCFG rule "PP NP-NN" frequently occurred in truthful reviews such as " . . . the spa is a closet [**with a massage table**]..." We conceive that for deceptive reviews, the reviewers wrote some imaginative information about the hotel and thus used past tense verbs. However, for truthful reviews, the reviewers wrote about a real experience at the hotel and thus used the present tense and noun phrases to make the review more specific and with more details.

We also used a Wilcoxon signed-rank test to investigate the statistical difference in the top 100 PCFG rules between different categories. Table 4 shows the significance level of each pair of categories. It can be seen that there is no difference in PCFG rules between positive and negative reviews but a significant difference between deceptive and truthful reviews.

Table 4. A Wilcoxon signed-rank test comparing the paired set of selected top 100 PCFG rules in each category of hotel reviews.

p-Value	PCFG N.D.	PCFG N.T.	PCFG P. D.	PCFG P. T.
PCFG N.D.	-	0.05**	0.28	0.05**
PCFG N.T.	0.05**	-	0.05**	0.31
PCFG P. D	0.28	0.29	-	0.05**

**: strong evidence of significance of rule difference.

3.2. The CoSpa Algorithm

The CoSpa algorithm, based on co-training and used to identify deceptive reviews, is shown in Algorithm 1. With Line 1 to Line 2, we use the labeled reviews L to train the base SVM classifier with two independent views on the reviews as lexical terms and PCFG rules. Then, with Line 4 to Line 8, we use the trained SVM classifiers to classify the unlabeled reviews in test set U' and selected the $2n + 2p$ classified reviews from U' to augment the training set L. Next, with Line 9, we fetch $2n + 2p$ reviews from the unlabeled set U to complement the test set U'. Finally, with Line 10, the base classifiers are retrained using the augmented training set L.

Algorithm 1. The CoSpa algorithm without details on selecting classified unlabeled reviews in the co-training process.

Input:

L—a set of n_L labelled reviews as deceptive or truthful;
U—a set of n_U unlabeled reviews;
X_1—the feature set of terms derived from the words in reviews;
X_2—the feature set of PCFG rules derived from deep syntax analysis on reviews;
K—the number of iterations;
$n_{U'}$—the number of reviews U' drawn from U;
n—the number of selected reviews which are classified as truthful;
p—the number of selected reviews which are classified as deceptive;
Y—the labels of reviews, *i.e.* $Y = \{deceptive, truthful\}$.

Output: two trained classifies $f_1^{(K)} : X_1 \rightarrow Y$ and $f_2^{(K)} : X_2 \rightarrow Y$.
Procedure:

1. $U' \leftarrow$ a set of $n_{U'}$ reviews randomly sampled from U;
2. Train a classifier $f_1^{(0)}$ on the X_1 view (*i.e.*, word view) of L, and a classifier $f_2^{(0)}$ on the X_2 view (*i.e.*, PCFG rule view) of L;
3. For $t = 1, ..., K$ iterations do
4. Use $f_1^{(t-1)}$ to classify the reviews in U';
5. Use $f_2^{(t-1)}$ to classify the reviews in U';
6. Select n reviews from U' classified by $f_1^{(t-1)}$ as truthful and p reviews from U' classified by $f_1^{(t-1)}$ as deceptive;
7. Select n reviews from U' classified by $f_2^{(t-1)}$ as truthful and p reviews from U' classified by $f_2^{(t-1)}$ as deceptive;
8. Add the selected $2n + 2p$ reviews to L and remove them from U';
9. Randomly choose $2n + 2p$ reviews from U and move them to U';
10. Retrain a classifier $f_1^{(t)}$ on the X_1 view (*i.e.*, word view) of L, and a classifier $f_2^{(t)}$ on the X_2 view (*i.e.*, PCFG rule view) of L;
11. End for.

A problem with the CoSpa algorithm is how to select the $2n + 2p$ reviews from U' to retrain the classifiers $f_1^{(K)} : X_1 \rightarrow Y$ and $f_2^{(K)} : X_2 \rightarrow Y$. That is to say, how to augment the training set L to

update $f_1^{(K)}$ and $f_1^{(K)}$ in each iteration. To tackle this problem, we develop two strategies, CoSpa-C and CoSpa-U, to select the classified reviews in U' of each iteration purposely. The CoSpa-C strategy selects the most confidently classified reviews by $f_1^{(K)}$ and $f_2^{(K)}$ to augment training set L. The confidence of classified reviews were described in Section 2.2 by using the distance of the data point of the review to the hyperplane of the trained SVM of the time. The CoSpa-U strategy adopts uniform sampling of classified reviews in U'. After sorting the reviews in U' by their confidences given by each classifier, we randomly select n and p reviews according to their rankings in U'.

The details of CoSpa-C algorithm are shown in Algorithm 2. With Line 2 to Line 5, we compute the values of decision functions of $f_1^{(K)}$ and $f_2^{(K)}$ for the test review x'. If $f_1^{(K)}(x')$ is larger than zero, then the test review x' is classified to positive class in the view $X_1 \rightarrow Y$. Otherwise, the test review x' is classified to negative class in the view $X_1 \rightarrow Y$. By analogy, if $f_2^{(K)}(x')$ is larger than zero, then the test review x' was classified to positive class in the view $X_2 \rightarrow Y$. Otherwise, the test review x' is classified to negative class in the view $X_2 \rightarrow Y$. With Line 6 to Line 9, we sort the test reviews x' in U' in descending order by the confidence. Note that we have two sorted lists for the same test reviews x' in U'. One is sorted by the confidence given by $f_1^{(K)}$ and the other is sorted by the confidence given by $f_2^{(K)}$. It is possible that some test reviews are sorted at top positions by both decision functions. In this case, we add these test reviews to the training set L only once. That is to say, the number of test reviews appended to L is smaller than $2n + 2p$.

Algorithm 2. The CoSpa-C strategy.

Input:

$f_1^{(t)}$—$X_1 \rightarrow Y$ a decision function learned by support vector machine as ($\omega_1^{(t)}$,$b_1^{(t)}$) the trained classifier to label unlabeled reviews as truthful or deceptive on the X_1 view;

$f_2^{(t)}$—$X_2 \rightarrow Y$ a decision function learned by support vector machine as ($\omega_2^{(t)}$,$b_2^{(t)}$) the trained classifier to label unlabeled reviews as truthful or deceptive on the X_2 view ;

U'—a set of $n_{U'}$ unlabeled reviews;

n—the number of selected reviews which are classified as truthful;

p—the number of selected reviews which are classified as deceptive;

Y—the labels of reviews, *i.e.*, $Y = \{deceptive,\ truthful\}$.

Output: L'—a set of $2n + 2p$ reviews with highest confidence.
Procedure:
For each review x' in U' do;

Compute $\omega_1^{(t)}x' + b_1^{(t)}$ and $\omega_2^{(t)}x' + b_2^{(t)}$;
If $\omega_1^{(t)}x' + b_1^{(t)} < 0$, then add x' to L_1^-; else add $x\prime$ to L_1^+;
If $\omega_2^{(t)}x' + b_2^{(t)} < 0$, then add x' to L_2^-; else add x' to L_2^+;

End for
Sort the reviews in L_1^- and L_1^+ respectively in descending order according to the absolute value of $\omega_1^{(t)}x\prime + b_1^{(t)}$;
Sort the reviews in L_2^- and L_2^+ respectively in descending order according to the absolute value of $\omega_2^{(t)}x' + b_2^{(t)}$;
Add the first n reviews in L_1^- and p reviews from L_1^+ to L' and remove them from U';
Add the first n reviews in L_2^- and p reviews from L_2^+ to L' and remove them from U';
Return L'.

The details of the CoSpa-U algorithm are shown in Algorithm 3. With Line 2 to Line 7, we compute the values of decision functions of $f_1^{(K)}$ and $f_2^{(K)}$ for the test review x' and sort the test reviews by the confidence. With Line 8 to Line 11, we randomly sample n position from the list L_1^- sorted by the view $X_1 \rightarrow Y$ with negative labels and add these n test reviews to the appended set L'. With Line 12 to Line 15, we randomly sample n position from the list L_2^- sorted by the view $X_2 \rightarrow Y$ with negative labels and add these n test reviews to the appended set L'. With Line 16 to Line 19, we randomly sample p position from the list L_1^+ sorted by the view $X_1 \rightarrow Y$ with positive labels and add these p test reviews to the appended set L'. With Line 20 to Line 23, we randomly sample p position from the list L_2^+ sorted by the view $X_2 \rightarrow Y$ with positive labels and add these p test reviews to the appended set L'.

Algorithm 3. The CoSpa-U strategy.

Input:

$f_1^{(t)}$—$X_1 \rightarrow Y$ a decision function learned by support vector machine as ($\omega_1^{(t)}$,$b_1^{(t)}$) the trained classifier to label unlabeled reviews as truthful or deceptive on the X_1 view;

$f_2^{(t)}$—$X_2 \rightarrow Y$ a decision function learned by support vector machine as ($\omega_2^{(t)}$,$b_2^{(t)}$) the trained classifier to label unlabeled reviews as truthful or deceptive on the X_2 view;

U'—a set of $n_{U'}$ unlabeled reviews;

n—the number of selected reviews which are classified as truthful;

p—the number of selected reviews which are classified as deceptive;

Y—the labels of reviews, *i.e.* $Y = \{deceptive, truthful\}$.

Output: L'—a set of $2n + 2p$ randomly sampled reviews.
Procedure:

1. For each review x' in U' do;
2. Compute $\omega_1^{(t)} x' + b_1^{(t)}$ and $\omega_2^{(t)} x' + b_2^{(t)}$;
3. If $\omega_1^{(t)} x' + b_1^{(t)}$<0, then add x' to L_1^-; else add x' to L_1^+;
4. If $\omega_2^{(t)} x' + b_2^{(t)}$<0, then add x' to L_2^-; else add x' to L_2^+;
5. End for.
6. Sort the reviews in L_1^- and L_1^+ respectively in descending order according to the absolute value of $\omega_1^{(t)} x' + b_1^{(t)}$;
7. Sort the reviews in L_2^- and L_2^+ respectively in descending order according to the absolute value of $\omega_2^{(t)} x' + b_2^{(t)}$;
8. For $i = 1, ..., n$
9. $r[i] = \left\lfloor Math.Random() * sizeOf(L_1^-) \right\rfloor + 1$;
10. End for.
11. Add all $r[i]$th reviews in L_1^- to L' and remove them from U';
12. For $i = 1, ..., n$
13. $r[i] = \left\lfloor Math.Random() * sizeOf(L_2^-) \right\rfloor + 1$;
14. End for.
15. Add all $r[i]$th reviews in L_2^- to L' and remove them from U';
16. For $i = 1, ..., p$
17. $r[i] = \left\lfloor Math.Random() * sizeOf(L_1^+) \right\rfloor + 1$;
18. End for.
19. Add all $r[i]$th reviews in L_1^+ to L' and remove them from U';
20. For $i = 1, ..., p$
21. $r[i] = \left\lfloor Math.Random() * sizeOf(L_2^+) \right\rfloor + 1$;
22. End for.
23. Add all $r[i]$th review in L_2^+ to L' and remove them from U';
24. Return L'.

4. Experiments

4.1. The Datasets

We use two datasets in the experiments to examine the performances of the CoSpa algorithm in spam review identification. The one is the spam dataset from Myle Ott *et al.* [5] and the other is the deception dataset from Li *et al.* [30]. For each review, we conducted part-of-speech analysis, stop-word elimination, and stemming and PCFG analysis. The part of speech of an English word is determined by the Stanford POS Tagger (english-bidirectional-distsim.tagger), which is a probabilistic parts-of-speech tagger and can be downloaded freely online [31]. We obtained the stop-words from USPTO (United States Patent and Trademark Office) patent full-text and image database [32]. The porter stemming algorithm is used to produce an individual word stem [33]. For each review, we extracted all of its sentences using the sentence boundary determination method described in [34]. Finally, we use the Stanford Parser [25] to produce the parse tree for each sentence. Tables 5 and 6 show basic information about reviews in the spam dataset and the deception dataset, respectively. Although the deception dataset also has a hotel category, we do not use this category in the experiment because the data in the

category is actually a duplicate of that in the spam dataset plus deceptive reviews by experts and the focus of the paper is not to identify deceptive reviews from experts but from ordinary people.

Table 5. The basic information about reviews in the spam dataset.

Polarity	Category	# of Hotels	# of Reviews	# of Sentences
Positive	Deceptive_from_MTurk	20	400	3043
	Truthful_from_Web	20	400	3480
Negative	Deceptive_from_MTurk	20	400	4149
	Truthful_from_Web	20	400	4483

Table 6. The basic information about reviews in the deception dataset.

Subject	Category	# of reviews	# of Sentences
doctor	deceptive_MTurk	356	2369
	truthful	200	1151
restaurant	Deceptive_MTurk	201	1827
	truthful	200	1892

The spam dataset contains 7677 lexical terms (words) and 43,519 PCFG rules including 3955 type 1 rules, 15,489 type 2 rules, 6830 type 3 rules, and 17,245 type 4 rules. The deception datasets contains 5160 terms (words) and 43,511 PCFG rules, including 3953 type 1 rules, 15,483 type 2 rules, 6830 type 3 rules, and 17,245 type 4 rules. Because Stanford POS Tagger adopts the Penn Treebank tagset [35] for sentence tagging, we obtain 45 POS tags for each dataset. For the base SVM classifier, the linear kernel $(u * v)^1$ is used to train the classification model because it has proved superior to non-linear kernels in text categorization [36].

4.2. Experimental Setup

We see from Algorithm 1 that the CoSpa algorithm has three parameters to be tuned to investigate its performance, *i.e.*, the number of iterations, K; the number of selected reviews classified as truthful, n; and the number of selected reviews classified as deceptive, p. We learned from Liu [37] and Hong [38] that the proportion of selected negative reviews to selected positive reviews in co-training should be equal to the proportion of negative reviews to positive reviews in the whole dataset. In our datasets, we see from Tables 5 and 6 that the number of truthful reviews is equal to the number of deceptive reviews in three out of four subsets, with the exception of the subset with the subject "doctor". To simplify, in the following experiments, we randomly select 200 deceptive reviews in the subset with subject "doctor" and assign n as equal to p.

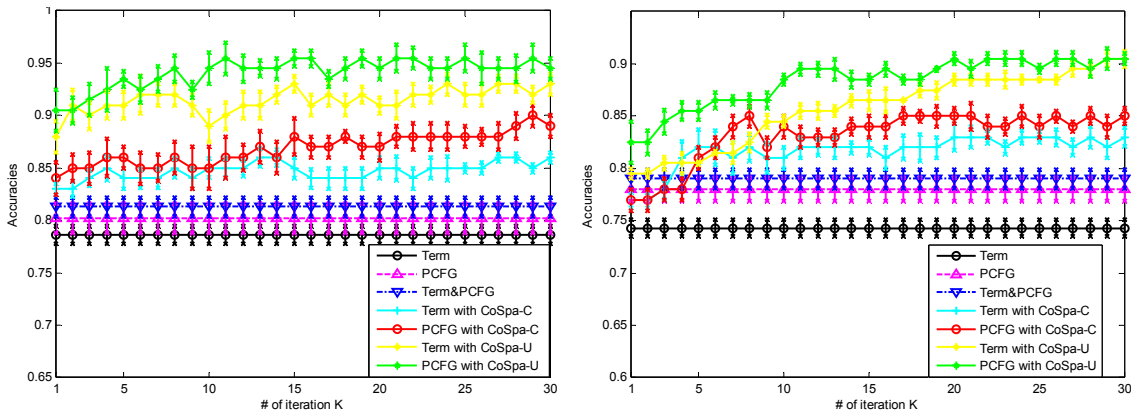
Thus, we devise two types of experiments to examine the performance of the CoSpa algorithm using the above two datasets. The first one is to tune the number of iterations K while fixing n and p. The second one is to tune the parameter n and p while fixing K. In the first experiment, we compare the CoSpa algorithm with state-of-the-art techniques in spam review identification including SVM with lexical terms (Term) proposed by Ott *et al.* [5], SVM with PCFG rules (PCFG) and POS tags proposed by Feng *et al.* [7], and SVM with lexical terms and PCFG rules (Term&PCFG) proposed by Feng *et al.* [7], as the baseline methods for comparison. We set $n = p = 3$ for the spam dataset and $n = p = 4$ for the deception dataset and vary the number of iteration K from 1 to 30. A five-fold cross-validation is used to average the performance. That is to say, we split the spam dataset into 320 reviews for training and 80 reviews for the test using five runs. In each run, we randomly select 5% of the data (16 reviews) for classifier training and 15% of the data (48 reviews) for the test to stimulate the scenario of not enough labeled data from the training split and the test split, respectively.

In the second experiment, we set $K = 12$ for the spam dataset because it has five runs and each run contains 320 reviews of four hotels for training. The accuracies of the compared algorithms are

averaged over the five runs. We also use 5% of the data (16 reviews) for classifier training and 15% of the data (48 reviews) for testing. Thus for each fold, there are only 256 (320−64) reviews remaining in U' to be appended to the training set. If n is set as 5, that means we select 10 reviews for the negative class and 10 reviews for the positive class to augment the training set. Thus, we can set the maximum number of iteration K as 12 (*i.e.*, $\lfloor 256/20 \rfloor$). By analogy, we set maximum K as 15 and maximum n as 7 for the deception dataset.

4.3. Experimental Results

Figure 2 shows the performances of the CoSpa algorithm with different number of iterations K compared with the state-of-the-art techniques in spam review identification on the spam dataset (**a**) and the deception dataset (**b**). We can see from Figure 2 that, without co-training, on the spam dataset, using the feature set by combining word terms and PCFG rules produced the best performance at 0.793. The feature set with PCFG rules produced better performance at 0.776 than that with purely word terms at 0.775. For the deception dataset, we see that using the feature set by combining word terms with PCFG rules produced the best performance at 0.818. Meanwhile, the feature set with PCFG rules produced better performance at 0.806 than that with purely word terms at 0.780. This outcome is consistent with that of Feng and Hirst [10], who used PCFG rules as a complementary feature set for terms to improve the performance of spam review identification. However, we here find that using PCFG rules alone can also produce at least comparable performance to that of terms alone. This finding also provides us with an incentive to use co-training in two views, the lexical term feature set and the PCFG rule feature set.



(**a**) The spam dataset, $n = p = 3$. (**b**) The deception dataset, $n = p = 4$.

Figure 2. Performances of the CoSpa algorithm with different K compared with the baseline methods.

To better illustrate the effectiveness of each method in Figure 2, the classic non-parameter Mann–Whitney U test [29] is employed to examine the statistical significance of performance difference. Table 7 demonstrates the results of the Mann–Whitney U test for the performance of the CoSpa algorithm and other baseline methods on accuracies of spam review identification in the spam and deception datasets. We use the accuracies of each method performing on all five runs for comparison. The following codification of the p-value in ranges was used: ">>" ("<<") means that the p-value is lesser than or equal to 0.01, indicating strong evidence that the target method outperforms the compared method; "<" (">") means that the p-value is bigger than 0.01 and minor or equal to 0.05, indicating weak evidence that the target method outperforms the compared baseline method; "~" means that the p-value is greater than 0.05, indicating that the compared methods do not have significant differences in performances.

Table 7. The performances on accuracies of the CoSpa algorithm compared with the state-of-the-art methods on spam review identification in the spam and deception datasets.

Dataset	Method Pair	$K = 1$	$K = 5$	$K = 10$	$K = 15$	$K = 20$	$K = 25$	$K = 30$
Spam	Term with CoSpa-C *vs.* Term	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-C *vs.* PCFG	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-C *vs.* Term&PCFG	>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-C *vs.* PCFG with CoSpa-C	<	<	~	<<	<<	<<	<<
	Term with CoSpa-U *vs.* Term	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-U *vs.* PCFG	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-U *vs.* Term&PCFG	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-U *vs.* PCFG with CoSpa-U	<	<	<<	<<	<<	<<	<<
	PCFG with CoSpa-C *vs.* PCFG with CoSpa-U	<<	<<	<<	<<	<<	<<	<<
Deception	Term with CoSpa-C *vs.* Term	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-C *vs.* PCFG	<	~	>>	>>	>>	>>	>>
	Term with CoSpa-C *vs.* Term&PCFG	<<	~	>	>>	>>	>>	>>
	Term with CoSpa-C *vs.* PCFG with CoSpa-C	>	>>	~	<<	<<	<<	<<
	Term with CoSpa-U *vs.* Term	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-U *vs.* PCFG	>>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-U *vs.* Term&PCFG	>	>>	>>	>>	>>	>>	>>
	Term with CoSpa-U *vs.* PCFG with CoSpa-U	<	<	<<	<<	<<	<<	~
	PCFG with CoSpa-C *vs.* PCFG with CoSpa-U	<<	<<	<<	<<	<<	<<	<<

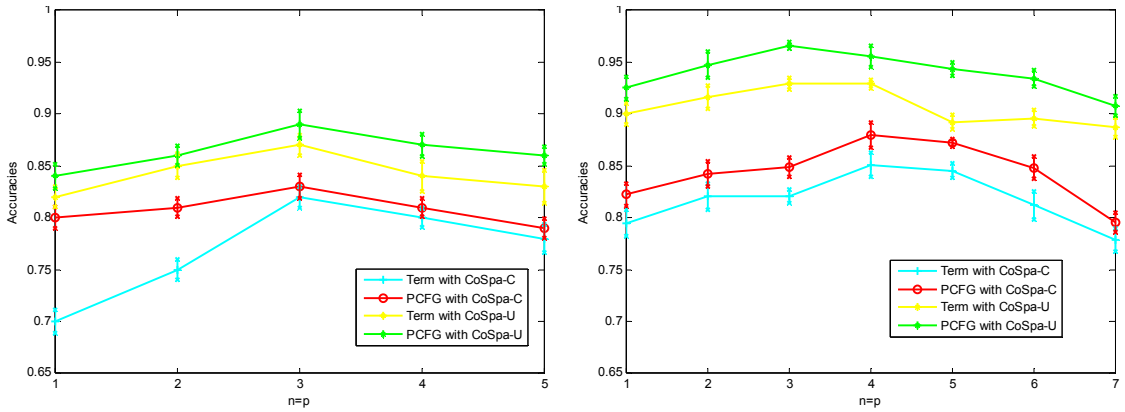
With co-training, firstly, we see that on the whole, the CoSpam algorithm produced better performance than those classifiers without co-training. On both the spam and deception datasets, we see that CoSpam-U algorithm produced better performance than the Co-Spam-C algorithm. We calculated that the CoSpam-U strategy has produced on average 5.68% improvements in accuracy in comparison with the CoSpa-C strategy on the spam dataset. On the deception dataset, the average improvement is 6.73% on accuracy. When using SVM as the base classifier in co-training, those data points with higher confidence appended into training set L are not necessarily the "optimal" choices [22] because those confidently classified reviews are farthest to the hyperplane of SVM with a small probability of updating the support vectors in the next iteration. However, when the uniformly random strategy is adopted, those data points, which are near the hyperplane of the time and with a large probability of updating the support vectors in the next iteration, are also appended into the training data set L. Note that despite that, noise or outlier data points will also be appended into the training data set in the uniform random strategy. However, because the accuracy of the current iteration is more than 80%, we can deduce that more "normal" data points were appended in to the training dataset L than noise or outlier data points. We admit that, at this time, we do not have a better strategy in selecting data points from U' to be appended into the training set L. On the one hand, we need to make sure that the appended data points are not noisy instances or outliers. On the other hand, we also need to make sure that those appended data points will improve the performance of the SVM classifier in the next iteration by updating its support vectors.

Secondly, we see that at the starting iterations, the performances of both the CoSpa-C and CoSpa-U strategies improve gradually. However, after about 10 to 15 iterations, the performances of the CoSpa-C and CoSpa-U strategies become stable. During the first iterations, the appended data points from U' to L provides some new knowledge for the SVM classifier to learn. However, after a number of iterations, the knowledge inherent in the remaining data points in U is similar to the knowledge learned by the current SVM classifier, thus making it stable.

Thirdly, we see that for both the CoSpa-C and CoSpa-U algorithms, the feature set of PCFG rules outperforms the feature set of word terms. This outcome can be attributed to the fact that initially, without co-training, the performance of PCFG rules is slightly better than that of word terms. Thus, we argue that the initial performance of a base classifier is crucial to the performance of co-training exhibited in the following iterations.

Figure 3 shows the performance of the CoSpa algorithm when setting a different number of selected reviews classified as truthful n and deceptive p. Table 8 also shows the significance test by

Mann–Whitney U test on accuracies between the compared methods in identifying spam reviews using the same codification as used in Table 7. We can see from Figure 3 and Table 8 that when we vary the number of selected reviews to augment the training data, the accuracies of both Spam-C and Spam U increased firstly and then decreased after a critical number. For the spam dataset, the critical number is 3 and for the deception dataset the critical number is 4, in that the size of the deception dataset for training is larger than that of the spam dataset. At the beginning stage of co-training, the performance of both CoSpa-C and CoSpa-U algorithms is boosted because those selected reviews added some new knowledge to retrain the classifier. However, when the number of selected reviews became larger than the critical numbers, most knowledge that can be learned from the dataset has already been captured by the existing training data. Thus those appended data after the critical number become redundant and are not necessary to boost the performance of the classifier. We also see from Figure 3 that the CoSpa-U strategy outperforms the CoSpa-C strategy with both the term set and the PCFG feature set. Moreover, in both datasets, the PCFG feature set outperforms the term feature set. The same explanation for Figure 2 can be used here.



(**a**) The spam dataset, $K = 12$. (**b**) The deception dataset, $K = 15$.

Figure 3. Performance of the CoSpa algorithm with different $n = p$ compared with different strategies.

Table 8. The performance accuracies of the CoSpa algorithm with different strategies for selecting classified unlabeled reviews.

Dataset	Method Pair	$n = p = 1$	$n = p = 2$	$n = p = 3$	$n = p = 4$	$n = p = 5$	$n = p = 6$	$n = p = 7$
Spam	Term with CoSpa-C *vs.* Term with CoSpa-U	<<	<<	<<	<<	<<	-	-
	Term with CoSpa-C *vs.* PCFG with CoSpa-C	<<	<<	<	<	<	-	-
	Term with CoSpa-C *vs.* PCFG with CoSpa-U	>	>>	>>	>>	>>	-	-
	PCFG with CoSpa-C *vs.* Term with CoSpa-U	<<	<<	<<	<<	<<	-	-
	PCFG with CoSpa-C *vs.* PCFG with CoSpa-U	<<	<<	<<	<<	<<	-	-
	Term with CoSpa-U *vs.* PCFG with CoSpa-U	<<	<	<<	<<	<<	-	-
Deception	Term with CoSpa-C *vs.* Term with CoSpa-U	<<	<<	<<	<<	<<	<<	<<
	Term with CoSpa-C *vs.* PCFG with CoSpa-C	<<	<<	<<	<<	<<	<<	<<
	Term with CoSpa-C *vs.* PCFG with CoSpa-U	<<	<<	<<	<<	<<	<<	<<
	PCFG with CoSpa-C *vs.* Term with CoSpa-U	<<	<<	<<	<<	<<	<<	<<
	PCFG with CoSpa-C *vs.* PCFG with CoSpa-U	<<	<<	<<	<<	<<	<<	<<
	Term with CoSpa-U *vs.* PCFG with CoSpa-U	<<	<<	<<	<<	<<	<<	<<

5. Concluding Remarks

In this paper, we propose co-training for spam review detection by using two "views" for each review: one is the shallow lexical terms derived from the textual content and the other is the PCFG rules derived from a deep syntax analysis of the review. The three main contributions of the paper can be summarized as follows:

- Firstly, using the spam dataset, we came up with the observation of difference on lexical terms and PCFG rules distributed in deceptive and truthful reviews.
- Secondly, we proposed the CoSpa algorithm based on a support vector machine to implement co-training using two representations for each review, lexical terms and PCFG rules. Further, we proposed two strategies, Co-Spa-C and CoSpa-U, to select informative unlabeled data to improve the performance of the CoSpa algorithm.
- Thirdly, we conduct experiments on the spam dataset and the deception dataset to propose the CoSpa algorithm and other baseline methods. Experimental results demonstrate that both the CoSpa-C and CoSpa-U strategies outperform a traditional SVM with lexical terms or PCFG rules in spam review detection. The CoSpa-U strategy outperforms the CoSpa-C strategy in spam review identification. The representation using PCFG rules outperforms the representation using lexical terms. We explain the experiment outcome in the paper.

Inspired by the outcome of the experiments, in the future, on the one hand, we will introduce data editing techniques [22] to the CoSpa algorithm to select the most informative unlabeled data for retraining the classifier to improve spam review identification to the greatest extent. On the other hand, we will study the sentiment analysis [39] of the reviews, especially on negation detection [40], and combine them to enhance spam review identification.

Acknowledgments: This work is supported by the National Natural Science Foundation of China under Grant Nos 71101138, 61379046 and 61432001; the Fundamental Research Funds for the Central Universities in BUCT.

Author Contributions: Wen Zhang and Taketoshi Yoshida conceived and designed the experiments; Wen Zhang performed the experiments; Wen Zhang and Chaoqi Bu analyzed the data; Siguang Zhang contributed analysis tools; Wen Zhang wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aljukhadar, M.; Senecal, S. The user multifaceted expertise: Divergent effects of the website versus e-commerce expertise. *Int. J. Inf. Manag.* **2016**, *36*, 322–332. [CrossRef]
2. Xiang, Z.; Magnini, V.P.; Fesenmaier, D.R. Information technology and consumer behavior in travel and tourism: Insights from travel planning using the Internet. *J. Retail. Consum. Serv.* **2015**, *22*, 244–249. [CrossRef]
3. Zhang, W.; Wang, S.; Wang, Q. KSAP: An approach to bug report assignment using KNN search and heterogeneous proximity. *Inf. Softw. Technol.* **2016**, *70*, 68–84. [CrossRef]
4. Li, H.; Chen, Z.; Liu, B.; Wei, X.; Shao, J. Spotting Fake Reviews via Collective Positive-Unlabeled Learning. In Proceedings of 2014 IEEE International Conference on Data Mining (ICDM), Shenzhen, China, 14–17 December 2014.
5. Ott, M.; Choi, Y.; Cardie, C.; Hancock, J.T. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011; pp. 309–319.
6. Pennebaker, J.W.; Chung, C.K.; Ireland, M.; Gonzales, A.; Booth, R.J. *The Development and Psychometric Properties of LIWC2007*; LIWC.net: Austin, TX, USA, 2007.
7. Feng, S.; Banerjee, R.; Choi, Y. Syntactic Stylometry for Deception Detection. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Korea, 8–14 July 2012; pp. 171–175.
8. Feng, V.W.; Hirst, G. Detecting deceptive opinions with profile compatibility. In Proceedings of the International Joint Conference on Natural Language Processing, Nagoya, Japan, 14–18 October 2013; pp. 338–346.
9. Zhou, L.; Shi, Y.; Zhang, D. A Statistical Language Modeling Approach to Online Deception Detection. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 1077–1081. [CrossRef]
10. Li, H.; Chen, Z.; Mukherjee, A.; Liu, B.; Shao, J. Analyzing and Detecting Opinion Spam on a Large scale Dataset via Temporal and Spatial Patterns. In Proceedings of The 9th International AAAI Conference on Web and Social Media (ICWSM-15), Oxford, UK, 26–29 May 2015.
11. Jindal, N.; Liu, B. Opinion Spam and Analysis. In Proceedings of 2008 International Conference on Web Search and Data Mining (WSDM'08), Palo Alto, CA, USA, 11–12 February 2008.

12. Li, F.; Huang, M.; Yang, Y.; Zhu, X. Learning to Identifying Review Spam. In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'11), Barcelona, Spain, 16–22 July 2011.
13. A Statistical Analysis of 1.2 Million Amazon Reviews. Available online: http://minimaxir.com/2014/06/reviewing-reviews/ (accessed on 1 March 2016).
14. Fact Sheet of Tripadvisor. Available online: http://www.tripadvisor.com/PressCenter-c4-Fact_Sheet.html (accessed on 1 March 2016).
15. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning theory (COLT' 98), Madisson, WI, USA, 24–26 July 1998; pp. 92–100.
16. Heydari, A.; Tavakoli, M.; Salim, N.; Heydari, Z. Detection of review spam: A survey. *Expert Syst. Appl.* **2015**, *42*, 3634–3642. [CrossRef]
17. Fusilier, D.H.; Montes-y-Gómez, M.; Rosso, P.; Cabrera, R.G. Detecting positive and negative deceptive opinions using PU-learning. *Inf. Process. Manag.* **2015**, *51*, 433–443. [CrossRef]
18. Ben-David, S.; Lu, T.; Pal, D. Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning. In Proceedings of the 21st Annual Conference on Learning Theory, Helsinki, Finland, 9–12 July 2008; pp. 33–44.
19. Marc-A, K.; Tobias, S. Multi-Relational Learning, Text Mining, and Semi-Supervised Learning for Functional Genomics. *Mach. Learn.* **2014**, *57*, 61–81.
20. Wang, W.Y.; Thadani, K.; McKeown, K.R. Identifying Event Descriptions using Co-training with Online News Summaries. In Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 8–13 November 2011.
21. Mihalcea, R. Co-training and self-training for word sense disambiguation. In Proceedings of the 2nd Conference on Computational Natural Language Learning, Boston, MA, USA, 26–27 May 2004.
22. Du, J.; Ling, C.X.; Zhou, Z.H. When does co-training work in real data? *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 788–799. [CrossRef]
23. Liu, W.; Li, Y.; Tao, D.; Wang, Y. A general framework for co-training and its applications. *Neurocomputing* **2015**, *167*, 112–121. [CrossRef]
24. Collins, M. Probabilistic Context-Free Grammars (PCFGs). 2013. Available online: http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf (accessed on 26 February 2016).
25. Klein, D.; Manning, C.D. Accurate Unlexicalized Parsing. In Proceedings of the 41st Meeting of the Association for Computational Linguistics, Sapporo, Japan, 7–12 July 2003; pp. 423–430.
26. Wan, X. Co-training for cross-lingual sentiment classification. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Stroudsburg, PA, USA; 2009; pp. 235–243.
27. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer-Verlag: New York, NY, USA, 1995.
28. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.
29. Sidney, S. *Non-parametric Statistics for the Behavioral Sciences*; McGraw-Hill: New York, NY, USA, 1956; pp. 75–83.
30. Li, J.; Ott, M.; Cardie, C.; Hovy, E. Towards a General Rule for Identifying Deceptive Opinion Spam. In Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, MD, USA, 22–27 June 2014; pp. 1566–1576.
31. Stanford POS Tagger for English part-of-speech. Available online: http://nlp.stanford.edu/software/tagger.shtml (accessed on 1 March 2016).
32. USPTO stop words. Available online: http://ftp.uspto.gov/patft/help/stopword.htm (accessed on 1 March 2016).
33. Porter stemming algorithm. Available online: http://tartarus.org/martin/PorterStemmer/ (accessed on 1 March 2016).
34. Weiss, S.M.; Indurkhya, N.; Zhang, T.; Damerau, F. *Text Mining: Predictive Methods for Analyzing Unstructured Information*; Springer-Verlag: New York, NY, USA, 2004; pp. 36–37.
35. Penn Treebank Tag-set. Available online: http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html (accessed on 1 March 2016).

36. Zhang, W.; Yoshida, T.; Tang, X. Text classification based on multi-word with support vector machine. *Knowl.-Based Syst.* **2008**, *21*, 879–886. [CrossRef]
37. Liu, B.; Feng, J.; Liu, M.; Hu, H.; Wang, X. Predicting the quality of user-generated answers using co-training in community-based question answering portals. *Pattern Recognit. Lett.* **2015**, *58*, 29–34. [CrossRef]
38. Hong, Y.; Zhu, W. Spatial Co-Training for Semi-Supervised Image Classification. *Pattern Recognit. Lett.* **2015**, *63*, 59–65. [CrossRef]
39. Ravi, K.; Ravi, V. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowl.-Based Syst.* **2015**, *89*, 14–46. [CrossRef]
40. Xia, R.; Xu, F.; Yu, J.; Qi, Y. Erik Cambria: Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis. *Inf. Process. Manag.* **2016**, *52*, 36–45. [CrossRef]