Robustness of Word and Character N-gram Combinations in Detecting Deceptive and Truthful Opinions

AL HAFIZ AKBAR MAULANA SIAGIAN, Kumamoto University, Japan and Indonesian Institute of Sciences, Indonesia
MASAYOSHI ARITSUGI, Kumamoto University, Japan

Opinions in reviews about the quality of products or services can be important information for readers. Unfortunately, such opinions may include deceptive ones posted for some business reasons. To keep the opinions as a valuable and trusted source of information, we propose an approach to detecting deceptive and truthful opinions. Specifically, we explore the use of word and character n-gram combinations, function words, and word syntactic n-grams (word sn-grams) as features for classifiers to deal with this task. We also consider applying word correction to our utilized dataset. Our experiments show that classification results of using the word and character n-gram combination features could perform better than those of employing other features. Although the experiments indicate that applying the word correction might be insignificant, we note that the deceptive opinions tend to have a smaller number of error words than the truthful ones. To examine robustness of our features, we then perform cross-classification tests. Our latter experiments results suggest that using the word and character n-gram combination features could work well in detecting deceptive and truthful opinions. Interestingly, the latter experimental results also indicate that using the word sn-grams as combination features could give good performance.

CCS Concepts: • **Information systems** → **Spam detection**; *Data analytics*; • **Computing methodologies** → *Information extraction*; • **Applied computing** → Document management and text processing;

Additional Key Words and Phrases: Deceptive opinions, robustness, sn-grams, fake reviews, spell correction, word and character combinations

ACM Reference format:
Al Hafiz Akbar Maulana Siagian and Masayoshi Aritsugi. 2020. Robustness of Word and Character N-gram Combinations in Detecting Deceptive and Truthful Opinions. *J. Data and Information Quality* 12, 1, Article 5 (January 2020), 24 pages.
https://doi.org/10.1145/3349536

5

A. H. A. M. Siagian thanks the scholarship support from RISET-Pro (Research and Innovation in Science and Technology Project) KEMENRISTEKDIKTI (Ministry of Research, Technology and Higher Education of the Republic of Indonesia).
Authors' addresses: A. H. A. M. Siagian, Kumamoto University, Computer Science and Electrical Engineering, Graduate School of Science and Technology, 2-39-1 Kurokami, Chuo-ku, Kumamoto, Kumamoto, 860-8555, Japan, and Indonesian Institute of Sciences, Jl. Jenderal Gatot Subroto Kav.10, Jakarta, DKI Jakarta, 12710, Indonesia; email: alha002@dbms.cs.kumamoto-u.ac.jp; M. Aritsugi, Kumamoto University, Big Data Science and Technology, Faculty of Advanced Science and Technology, 2-39-1 Kurokami, Chuo-ku, Kumamoto, Kumamoto, 860-8555, Japan; email: aritsugi@cs.kumamoto-u.ac.jp.

1936-1955/2020/01-ART5 $15.00
https://doi.org/10.1145/3349536

1 INTRODUCTION

Usually, prospective customers read opinions in reviews of a product or service to learn the quality of it. According to Reference [31], 90% of the customers notice on such reviews to be one of their purchasing considerations. Indeed, 88% of them trust in those reviews to convince themselves [31]. However, due to business reasons, this habit is abused by generating deceptive opinions [41]. The deceptive opinions are mainly used to upgrade the sales or to discredit the competitor, which is known as positive and negative deceptive opinions, respectively [8]. Detecting deceptive and truthful opinions is a crucial effort to keep the reviews as a valuable and trusted source of information.

Since deceptive opinions look similar to truthful ones, to detect them is a challenging and complicated task. References [13–15] have initiated detecting deceptive and truthful opinions in reviews, as well as review spam analysis. They have concluded that it might not be easy to identify deceptive and truthful opinions manually. Moreover, Reference [11] has shown that employing a human to recognize the deceptive and truthful opinions could not work well. Interestingly, References [11, 24, 25] have indicated that using an automatic classification system could perform better than the human assessment to deal with this task. For this reason, we consider using a machine learning approach in this study. Particularly, we explore the use of word and character combinations, function words, and word syntactic n-grams as features in detecting deceptive and truthful opinions.

Employing word or character n-grams as features for classifiers has shown that it could obtain promising results to classify the deceptive and truthful opinions [2, 3, 7–9, 24, 25]. Based on their encouraging results, we are motivated to explore the feasibility of combining the word and character n-grams as features to deal with this task. We presume that by combining between the word and character n-grams, it would cover the lack of each of them. In other words, we expect that the word and character n-gram combinations would be useful features in detecting deceptive and truthful opinions.

Furthermore, using function words as features has been successful to improve the obtained results in text classification tasks [17, 18, 22, 40]. We thus adopt their idea to utilize the function words features in detecting deceptive and truthful opinions. By using the function words features, we expect that it could not only improve our classification performance but also be a useful hint to identify deceptive and truthful opinions, i.e., in terms of the number of function words. As in Reference [22], we consider combining the function words features with other utilized features in this study.

References [37–39] introduced syntactic[1] n-grams (sn-grams) as new features in text classifications tasks. Unlike traditional[2] n-grams, sn-grams could represent the linguistic information of text, i.e., based on the syntactic relation of words [37–39]. Since constructing sn-grams uses the syntactic relation, the attribute element of sn-grams is slightly different from that of traditional n-grams when $n > 1$, i.e., bigrams and trigrams. This construction behavior is able to form a more common[3] attribute element than that of the traditional n-grams [37–39]. Also, References [37–39] have shown that the obtained classification results of using the sn-gram features could outperform those of utilizing traditional ones. According to their successful work, we aim to use the sn-grams, that is, word sn-grams, as potential features in detecting deceptive and truthful opinions. Specifically, we explore the use of sn-bigrams and sn-trigrams when combined with our other employed features in this study. We expect that the particular characteristic of word sn-grams would give a meaningful contribution to detecting deceptive and truthful opinions.

[1]The neighbor element is assigned by following syntactic relation of parsing results.
[2]The neighbor element is based on its appearance in texts.
[3]The attribute element of features may be useful not only in a particular dataset but also in other datasets.

However, most studies in detecting deceptive and truthful opinions have focused only on proposing features for classifiers [4]. In this study, we consider applying word correction to our utilized dataset as well. Our motivation is due to the finding of References [12, 43]. Particularly, References [12, 43] could obtain better text classification results by applying a correction to their datasets. Our objective to do this is threefold. First, we would like to analyze the particular characteristic between deceptive and truthful opinions, that is, by observing the number of error words in each of them. Second, we intend to examine the robustness of our features when dealing with real conditions, i.e., with or without having error words in the reviews. Third, we aim to study the use of applying the word correction in this task.

We evaluate our work by utilizing a dataset that consists of deceptive and truthful opinions in positive and negative instances about Chicago hotel reviews [24, 25]. We then apply a word correction to that dataset. Our experiments seem to show that the obtained classification results of using the word and character n-gram combination features could outperform those of employing baseline features, i.e., the individual of word and character n-grams. These better-obtained results are in both conditions of the dataset, i.e., before and after applying the word correction. However, the experiments also indicate that using the word correction might be unnecessary when dealing with this task. Intriguingly, we find that the deceptive opinions appear to contain a less number of error words than the truthful ones.

To examine robustness of our features, we conduct cross-classification tests in this study. To do this, we use the Chicago hotel dataset as training sets in cross-hotel [19], cross-domain [20], and both positive and negative cross-type [23] classifications. Overall, our last experiments suggest that using the word and character n-gram combinations could be robust features in this task. The last experiments also indicate that using features combined with the word sn-grams could obtain good results in detecting deceptive and truthful opinions.

This article is essentially based on our previous work [32, 33]. Particularly, Reference [32] analyzed the use of word and character n-gram combinations as features in detecting deceptive and truthful opinions on hotel reviews [24, 25]. Meanwhile, Reference [33] exploited function words as features in classifying deceptive and truthful opinions on positive instances of several corpora [20, 23, 25]. However, our prior work mainly focused on evaluating the proposed features in intra-classification tests. Therefore, we aim to examine further these prior work in this study, that is, by utilizing word sn-gram features, applying word correction to the utilized dataset, and performing cross-classification tests.

Concretely, the main contributions of this article are as follows:

- We utilize further word and character n-gram combinations [32] and function words [33] as features in detecting deceptive and truthful opinions. We then exploit word sn-grams as potential features in this task. Particularly, we explore the word sn-gram features when incorporated with our other employed features in this study.
- We consider applying word correction in this study. Our consideration is not only to analyze the characteristic between deceptive and truthful opinions in terms of the number of error words but also to study the usefulness of doing the word correction in this task. Furthermore, this approach is to verify our prior conjecture [32] that using word and character n-gram combinations as features could perform well to detect deceptive and truthful opinions in the real case, i.e., having or no error words in reviews.
- We examine robustness of our features when employed in cross-classification tests. This examination is unlike our previous work [32, 33] that evaluated the performance of our features in detecting deceptive and truthful opinions on a particular dataset only.

The rest of this article is organized as follows. Section 2 presents some related work. Section 3 describes our employed method. Section 4 explains the dataset and experimental setup utilized in this study. Section 5 exhibits and discusses our obtained results. Section 6 provides our conclusion and future work.

2 RELATED WORK

Detecting deceptive and truthful opinions in reviews can be considered as text classification issues. Prior work has shown that employing text features, i.e., word or character n-grams, for classifiers could work well to deal with task [2, 3, 7–9, 24, 25]. The use of word n-grams as main features has shown that it could perform well in References [24, 25] work. Furthermore, References [7, 8] utilize the word n-gram features to evaluate their PU-learning method when dealing with this task. Following studies [2, 3, 9] have demonstrated that using character n-grams as features could work very well to classify deceptive and truthful opinions. However, the prior work has only focused on using the word or character n-gram features individually. We thus are encouraged to study the possibility of combining the word and character n-grams as features in this work.

Utilizing other text features such as function words has shown that it could improve classification results [17, 18, 22, 40]. The function words, e.g., conjunctions and prepositions, basically have no special meaning when used alone, but the use of them is necessary. In fact, the function words are more often used than other types of words [17]. For this reason, References [17, 18, 22, 40] have employed the function words features in their exhaustive work. Particularly, they identified that the function words could play an important role when used as features. As a result, they could obtain better classification results in the authorship attribution classification or native language identification tasks [17, 18, 22, 40]. Since the reviews probably consist of a lot of words, we consider using the function words features in detecting deceptive and truthful opinions.

Furthermore, instead of using traditional n-grams, syntactic n-grams (sn-grams) are proposed as new features in the classification tasks. Specifically, References [37–39] exhibited that the classification results of using sn-gram features could outperform those of employing traditional n-gram ones. To construct the sn-grams, References [37–39] have utilized syntactic trees obtained from the Stanford parser [5]. The syntactic trees are used to assign the neighbor element of the sn-grams. This construction behavior of sn-grams is different from that of the traditional n-grams, in which the neighbor element of the traditional n-grams is determined by its appearance in texts. Significantly, unlike the traditional n-grams, the sn-grams are not only able to reflect the linguistic information of text but also able to assign the attribute element of features in a more common order. Although there are significant differences between sn-grams and traditional n-grams, References [37–39] reported that the sn-grams could be used to tasks where the traditional n-grams are applied. Since the main object of References [37–39] was about authorship attribution task, we thus would like to explore the use of sn-gram features, i.e., word sn-grams, in detecting deceptive and truthful opinions.

However, applying a correction to the dataset can be an effective way to increase the classification results in text classification tasks. A recent study [43] has shown that by correcting error words in a social media dataset, i.e., Twitter and Formspring.me, it was able to improve the classifying accuracy. This better result corresponded to Reference [12] work that also obtained better classifying performance. In particular, Reference [12] applied the correction to Twitter corpora about Barack Obama and Microsoft. Although References [12, 43] could achieve a rising classifying accuracy in their work, their objectives were in sentiment analysis classification, i.e., not detecting deceptive and truthful opinions.

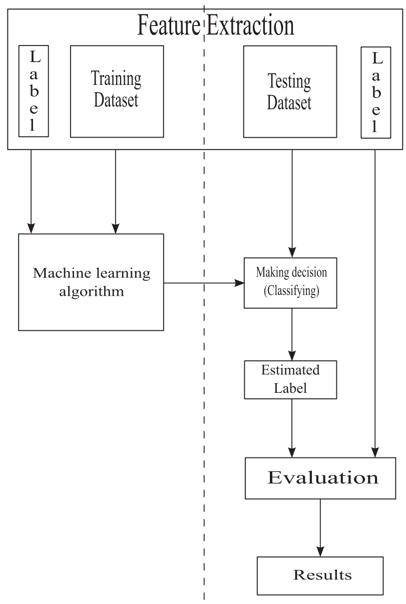

Fig. 1. The general overview of our method.

Several studies have performed cross-classification tests in detecting deceptive and truthful opinions [19, 20, 23]. Particularly, References [19, 20] conducted cross-hotel[4] and cross-domain[5] classifications to check the performance of classifiers in their work. Meanwhile, Reference [23] performed cross-type classifications to inspect the effectiveness of deceptive syntactic[6] reviews when used as training sets for classifying real-world[7] deceptive reviews. In contrast with References [19, 20, 23], our purpose to consider the cross-classification tests is to examine the robustness of our utilized features in this study. Moreover, References [19, 20, 23] have only focused on using the word n-gram features in their work.

3 METHOD

In this section, we explain our method and approach in this study. Basically, our method utilizes a supervised machine learning in detecting deceptive and truthful opinions. By using our proposed features, the machine learning learns from the provided training data to classify the testing data. Figure 1 represents a general overview of our method.

3.1 Word and Character N-gram Combinations

To begin with, we extract reviews in datasets to generate each of word and character n-gram features. We follow a traditional approach of the word and character n-grams by selecting n contiguous words or characters from texts in reviews.

At first, we create word n-gram features, i.e., unigrams, uni+bigrams, and uni+bi+trigrams. Then, we construct 4 and 5 grams of character n-gram features. This consideration is based on the study in Reference [9]. Particularly, they obtained the best classification results by using 5 and

[4]Chicago hotel dataset as training sets for classifying other cities hotel corpora.
[5]Chicago hotel dataset as training sets for classifying doctor and restaurant corpora.
[6]Deceptive reviews are generated by using Amazon Mechanical Turk (http://mturk.com).
[7]Deceptive reviews are obtained from filtered reviews on Yelp (https://www.yelp.com/).

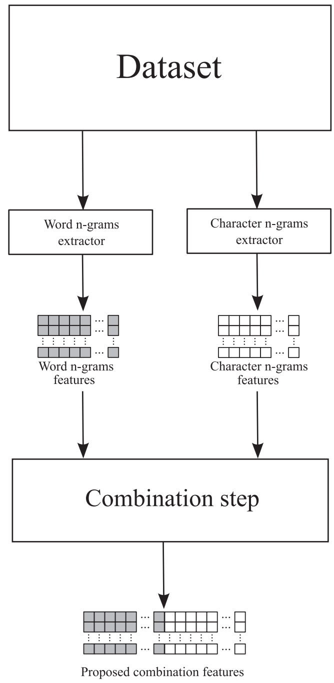

Fig. 2. Combination of word and character n-gram features.

4 grams of character n-gram features for detecting deceptive opinions in positive and negative instances, respectively.

As illustrated in Figure 2, we examine the feasibility of combining the word and character n-grams with respect to the results reported in Reference [9]. Since our proposal is to examine the feasibility of combining word and character n-grams as features, we incorporate them without any other considerations. In other words, we combine each one of the word n-grams with each one of the character n-grams to become new features at the combination step. Then, we provide these new features to create the training and testing datasets for the classifier. We present baseline features, i.e., the original features for each word and character n-grams, and our proposed combinations features in Table 1.

We assume that using the word and character n-gram combinations as features would provide reliable features. By doing the combinations, we expect they could cover the lack of each of word and character n-gram features for the machine learning process. Also, we expect these combinations features could deal with text problems, e.g., misspelled or mistyped, in the real case.

3.2 Function Words

Function words, such as conjunctions, determiners, prepositions, and pronouns, are types of words that do not have a particular meaning when used individually [17]. However, due to their typical characteristic, function words have shown a significant influence when used as features in authorship attribution classifications [17, 18, 40]. In particular, the use of function word features could help to identify author writing styles. Table 2 shows an example of function words.

Since our object in this study is opinions in reviews, we presume that the reviews may also contain function words. For this reason, we believe that using the function words as features would give a beneficial contribution to this study. Specifically, our aim to use function word features is

Table 1. The Baseline and Proposed Combinations Features

Baseline features	
Word n-grams	Character n-grams
Unigrams	4 char
Uni+Bigrams	5 char
Uni+Bi+Trigrams	

(a)

Proposed combinations features
Uni+4 char
Uni+5 char
Uni+Bi+4 char
Uni+Bi+5 char
Uni+Bi+Tri+4 char
Uni+Bi+Tri+5 char

(b)

Table 2. Example of Function Words

Function words			
Conjuctions	Determiners	Prepositions	Pronouns
When	The	On	They
Although	Much	At	He
While	Either	Between	One
That	More	Of	It

twofold. First, it can contribute to improving the classification results in detecting deceptive and truthful opinions. Second, it can be an indicator to identify deceptive and truthful opinions.

For our purpose, we use a set of 318 function words provided by Reference [26]. Our motivation to use this collection is with respect to the results in Reference [22]. Specifically, Reference [22] observed several collections of function words and found that by using Reference [26]'s function words list they could obtain the optimum classification results. For this reason, we employ that same function words collection in this study.

According to our prior study [33], utilizing the function word features only could not perform well in detecting deceptive and truthful opinions. Therefore, in this study, we focus on combining the function words features with other utilized features, i.e., word n-grams, character n-grams, and word and character n-gram combinations. To do this, we follow the same combination approach as well as depicted in Figure 2. Note that our prior work [33] has only focused on positive instances of deceptive and truthful opinions. We thus are motivated to examine further function words as combinations features in this study.

3.3 Word Syntactic N-grams (word sn-grams)

Syntactic n-grams (sn-grams) are a relatively new type of feature in text classification tasks. The main idea of sn-grams is to construct another type of n-gram features in a different manner, i.e., using syntactic trees [37–39]. The sn-grams are able to reflect the linguistic information of text. In

Table 3. Example of Stanford Parser Result

det(hotel-2, the-1)
nsubj(clean-4, hotel-2)
cop(clean-4, was-3)
root(ROOT-0, clean-4)
cc(clean-4, and-5)
conj(clean-4, tidy-6)

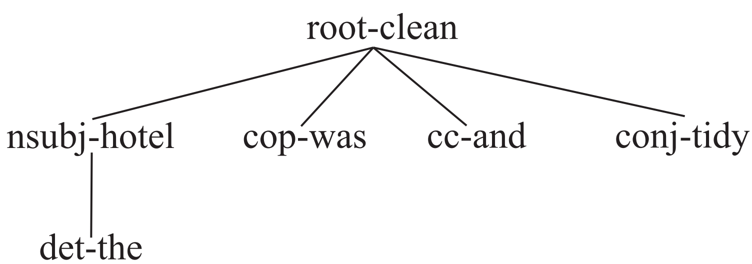

Fig. 3. Example of a syntactic tree. Words on the left side indicate syntactic relation tags, while the words on the right side represent words in the sentence.

addition, the attribute element of sn-grams may represent a more common structure than that of the traditional n-grams. In other words, these advantages of sn-grams cannot be acquired when using traditional n-grams as features. The sn-gram features have shown excellent performance when used to classify in authorship attribution tasks. Also, References [37–39] have reported that the sn-gram features could be used in the task wherever traditional n-gram features are applied. We thus are motivated to use sn-grams as potential features in this work. Specifically, we explore the use of word sn-gram features in detecting deceptive and truthful opinions.

Fundamentally, the concept of word sn-grams is similar to that of word traditional n-grams, especially when $n = 1$ (unigrams). The n represents the number of elements of both word sn-grams and traditional n-grams. However, constructing word sn-gram features with $n > 1$ is completely different from the traditional ones. To generate word sn-gram features, it utilizes a syntactic tree obtained after doing a parsing process. In particular, the neighbor elements of word sn-grams are assigned by following syntactic relations of that syntactic tree. This behavior is in contrast to the construction of word traditional n-grams, viz., to determine the neighbor elements of word traditional n-grams depends on the word's order in the text [37–39]. Based on this word sn-gram behavior, we expect it can give a meaningful contribution when employed in detecting deceptive and truthful opinions.

To create word sn-gram features, we employed the Stanford parser [5] and a supplied sn-gram builder program[8] [30, 37–39]. First, we used the Stanford parser for generating syntactic relations of each review in the dataset. We then utilized these syntactic relations as an input for the supplied sn-grams builder to produce the desired type of sn-gram features, that is, word sn-grams. We exhibit an example of the Stanford parser result using a sentence "the hotel was clean and tidy" from our utilized dataset in Table 3. Figure 3 shows the structure of a syntactic tree[9] obtained from the

[8] http://www.cic.ipn.mx/~sidorov/.
[9] We utilized the syntactic tree generator available on http://ironcreek.net/phpsyntaxtree/.

Table 4. Example of the Word Base of Sn-grams and Traditional N-grams

Word sn-grams	
Bigrams	Trigrams
clean was	clean hotel the
clean hotel	
hotel the	
clean and	
clean tidy	

(a)

Word traditional n-grams	
Bigrams	Trigrams
the hotel	the hotel was
hotel was	hotel was clean
was clean	was clean and
clean and	clean and tidy
and tidy	

(b)

Stanford parser result. According to References [37–39], when constructing word sn-grams with no bifurcations, the number of attributes of word sn-bigrams, i.e., $n = 2$, should be the same as the word traditional bigrams. However, the number of attributes of word sn-grams can be less or equal to word traditional n-grams when $n > 2$, e.g., trigrams and fourgrams. We display the obtained word bigrams and trigrams of sn-grams in Table 4(a) and those of traditional n-grams in Table 4(b).

Table 4(a) and (b) shows that the number of trigram attributes between word sn-grams and word traditional n-grams is completely different. The number of trigram attributes of word sn-grams is only one (Table 4(a)), while that of word conventional n-grams is four (Table 4(b)). This difference is because the number of trigram attributes of word traditional n-grams is generally based on the formula of the total number of words minus 2, i.e., $6 - 2 = 4$, while that of word sn-grams depends on the syntactic relations as portrayed in Figure 3, i.e., only has one trigram attribute relation. Another distinction found in Table 4(a) and (b) is the dissimilar word attribute of bigrams and trigrams between sn-grams and conventional n-grams. This distinction is because to assign the word attribute of sn-gram features also follows the syntactic relations.

We examine the use of word sn-bigrams and sn-trigrams in this study. In particular, we explore the ability of both word sn-bigram and sn-trigram features when incorporated with the word unigrams feature. We then consider combining the word sn-gram features with character n-gram or function word features in this study. To do this, we use the same method of performing combinations as pointed in Figure 2. Note that we use syntactic dependency-based and continuous[10] word sn-grams of References [37–39] only in this study.

3.4 Word Correction

According to References [12, 43], using word correction to a dataset could improve the classification results. Since their work was about sentiment analysis tasks, we are highly motivated to attest that possibility in this detecting deceptive and truthful opinions task. Therefore, we consider applying the word correction to our utilized dataset in this study.

In particular, our motivation to do this is threefold. First, we aim to investigate the typical characteristic between deceptive and truthful opinions in terms of the number of error words. Second, we need to check the effectiveness of our features when dealing with real situations, i.e., with or without having error words in the reviews. This second motivation is also to confirm our prior conjecture in Reference [32] that our combinations features could deal with error words such as misspelled or mistyped. Third, we intend to study the use of applying the word correction in this task. For these reasons, we take into consideration this approach in this study.

[10]Syntactic n-grams with no bifurcations.

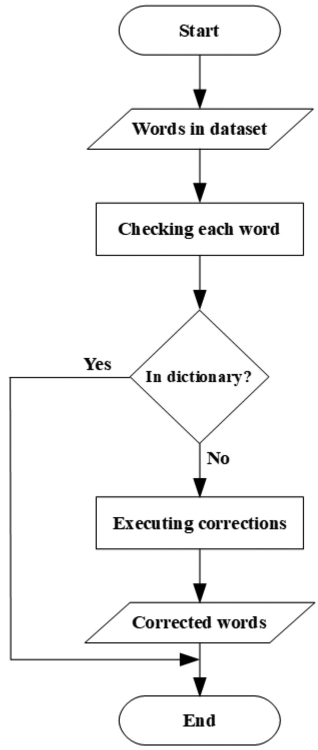

Fig. 4. The flowchart of processing word correction.

We describe our idea to process the word correction in the dataset as in Figure 4. For simplicity, we could summarize that thought as follows:

(1) Preparing a comprehensive dictionary to be used as a word database.
(2) Inspecting each word in the dataset whether the word is available or not in the dictionary.
(3) If the word is nonexistent in the word database, then executing the correction to that word automatically, i.e., without any human supervising or providing suggestions.

In this study, we borrow and apply a word corrector developed in Reference [43]. Specifically, Reference [43] has succeeded in developing an automatic application for correcting misspelled or mistyped words. In other words, this application conforms with our needs to do the word correction automatically. Note that due to our purpose in this study, we do not take into account the accuracy level of the replacement words in this study.

4 DATASET AND EXPERIMENTAL SETUP

4.1 Dataset

In this study, we utilize several datasets consisted of Chicago hotel reviews [24, 25],[11] doctor and restaurant reviews [20],[12] four cities hotel reviews [19, 25],[13] and Yelp's hotel reviews [23].[14] The

[11] http://myleott.com/op-spam.html.
[12] http://cs.stanford.edu/~bdlijiwei/deception_dataset.zip.
[13] http://web.stanford.edu/~jiweil/data/EMNLP_2013.zip.
[14] http://liu.cs.uic.edu/download/yelp_filter/.

method to construct the first three datasets is entirely identical. Specifically, they employed Amazon Mechanical Turk to create deceptive opinions, while they collected truthful reviews from Priceline, Expedia, Orbitz, TripAdvisor, Yelp, and Hotels.com [19, 20, 24, 25]. However, for the doctor and restaurant dataset, Reference [20] also gathered fake reviews by employing domain experts, i.e., doctor and restaurant employees, to write such opinions. In contrast, Reference [23] generated the real-life fake and truthful reviews by utilizing filtered and unfiltered reviews on Yelp site, viz., the Yelp's hotel reviews dataset.

In particular, the Chicago hotel dataset contains 1,600 labeled deceptive and truthful opinions about hotels in Chicago city for both positive and negative polarities. The doctor dataset has contents of 356 and 200 reviews labeled as positive deceptive and truthful opinions, respectively. The restaurant corpus has 200 deceptive and 200 truthful opinions of positive instances. The four cities dataset consists of 640 labeled positive both truthful and deceptive opinions obtained from two hotels in each of Los Angeles (LA), New York, Chicago, and Houston cities (each city has 80 deceptive and 80 truthful opinions). Then, the Yelp's hotel dataset has 780 filtered and 5,078 unfiltered of one- to five-star reviews that represent deceptive and truthful opinions, respectively.

For our experiment purposes, we allocate the use of those datasets as follows:

(1) We utilize only the Chicago hotel dataset [24, 25] for conducting the first, second, third, and fourth experiments in Section 5. Specifically, we use both positive and negative instances of deceptive and truthful opinions of the Chicago hotel dataset.
(2) Next, we employ positive instances both deceptive and truthful of the Chicago hotel dataset as training sets (800 reviews total) for classifying three types of testing sets as follows:
    - First, we use LA, Houston, and New York hotel reviews of the four cities dataset as testing set (480 reviews total) in a cross-hotel classification.
    - Second, we incorporate the doctor and restaurant reviews (956 reviews total) for using in a cross-domain classification.
    - Third, we make use of four- to five-star filtered and unfiltered reviews of Yelp's hotel dataset that represent positive instances (3,507 reviews total) to be classified in the positive cross-type classification, as well as Reference [23] did in their work.
(3) Finally, we use negative opinions both deceptive and truthful of Chicago hotel reviews as the training data (800 reviews total) for classifying one- to two-star filtered and unfiltered reviews of Yelp's hotel dataset that represent negative instances (1,260 reviews total) in the negative cross-type classification.

We present the statistic of these allocated datasets in Table 5.

4.2 Experimental Setup

In this work, we utilized WEKA tool [10] to perform all experiments. We also did some experimental settings to support and facilitate our experiments as follows:

(1) Tokenization preparation: We followed the dataset preprocessing method conducted by References [8, 9] that used alphabetical tokens only. In particular, we eliminated the punctuations and numerical signs. Then, we altered all alphabet letters into the lowercase.
(2) Classification algorithms: With respect to References [2, 3, 7–9, 19, 20, 23–25], we considered applying Naive Bayes (NB) and Support Vector Machine (SVM) classifiers in this study. We thus utilized the NB and SVM of WEKA's implementation [10] along with a binary weighting scheme to classify the deceptive and truthful opinions. To determine the best model of each NB and SVM, we examined various hyperparameters settings and

Table 5. The Statistic of Our Allocated Datasets for Experiments

Chicago hotel reviews			
Positive instances		Negative instances	
Deceptive	Truthful	Deceptive	Truthful
400	400	400	400

(a)

Doctor and restaurant reviews	
Positive instances	
Deceptive	Truthful
556	400

(b)

Four cities hotel reviews	
Positive instances	
Deceptive	Truthful
240	240

(c)

Yelp's hotel reviews			
Positive instances		Negative instances	
Deceptive	Truthful	Deceptive	Truthful
416	3091	284	976

(d)

then performed a 10-fold cross-validation to both NB and SVM models. Particularly, using the NB model with default hyperparameters or the SVM model with PolyKernel of the SMO and $C = 0.1$ could show the optimal classification performance in this study. However, the NB model seemed to consistently generate better classification results than the SVM one. Therefore, we show the obtained classification results of only the NB classifier in Section 5.

(3) Experimental measurement: We measured our proposed method by calculating Precision, Recall, and F-measure of each of deceptive and truthful opinions. To exhibit our experimental results, we used the macro average of F-measure of deceptive and truthful opinions. The Precision, Recall, and F-measure of each opinion are calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TN}{TN + FN}, \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TN}{TN + FP}, \tag{2}$$

$$F\text{-measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{3}$$

where:

TP = The number of instances correctly classified as deceptive opinions.

TN = The number of instances correctly classified as truthful opinions.

FP = The number of instances incorrectly classified as deceptive opinions.

FN = The number of instances incorrectly classified as truthful opinions.

(4) At last, regarding our purpose in this study, we did small customization on the word spell corrections application of Reference [43] by adding certain words, such as the name of hotels and roads, to its dictionary.

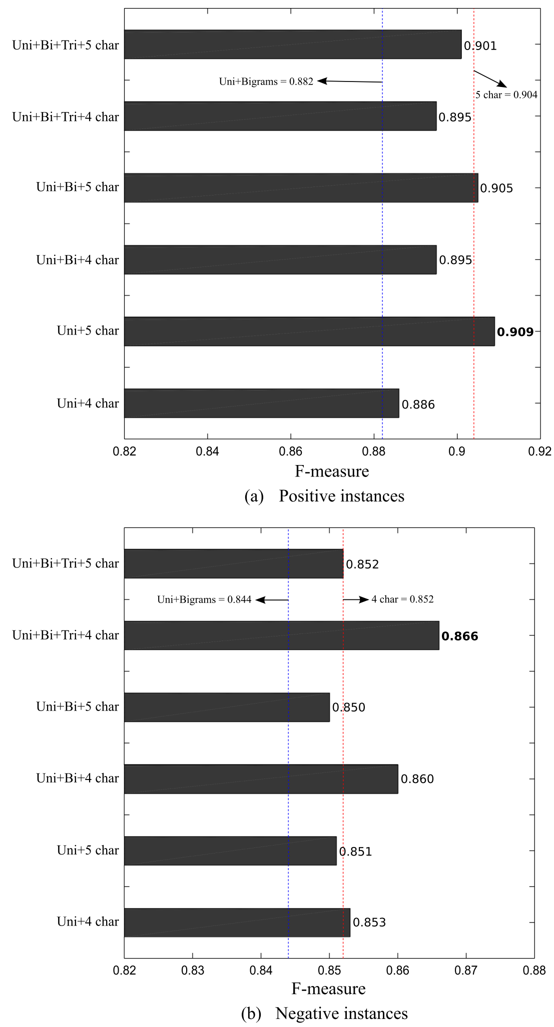

Fig. 5. Results obtained using the word and character n-gram combination features. Numbers in bold indicate the best performance for each polarity. The red dashed line stands for the best result of the baseline of character n-grams, while the blue dashed line represents the best result of the baseline of word n-grams.

5 EXPERIMENTS AND RESULTS

5.1 Experiment 1: Word and Character N-gram Combination Features

The main goal of this experiment is to analyze the feasibility of combining word and character n-grams as features in detecting deceptive and truthful opinions. In this study, we consider the individual of the word and character n-gram features as a baseline against our word and character n-gram combination features.

Figure 5 exhibits the obtained results of utilizing our word and character n-gram combination features. These results indicate that using word and character n-gram combination features could outperform both of the baseline results, i.e., using the individual of word and character n-gram features, for detecting deceptive and truthful opinions in both positive and negative instances. In particular, the classification of using the combination of Uni+5 char could obtain the best result in

Table 6. The Number of Words and Characters in Chicago Hotel Dataset

Chicago hotel dataset		
	Positive	Negative
Number of words	95,552	142,488
Number of characters	524,690	763,936

positive instances (Figure 5(a)), while that of using the combination of Uni+Bi+Tri+4 char could get the best result in negative instances (Figure 5(b)).

Regarding the difference of the best-obtained results between positive and negative instances in Figure 5, it might be due to the characteristic of their text contents, e.g., the number of words and characters utilized in each instance. The number of words and characters utilized in negative opinions was larger about 45% than that in positive ones (Table 6). This difference might be the cause of using the same features could not obtain similar good results for detecting deceptive and truthful opinions in both positive and negative instances. Indeed, this finding corresponds to Reference [9], which deduced that the employed vocabularies in negative opinions were quite larger than those in positive ones. Hence, their best-obtained classification results were also different between positive and negative instances. For this reason, to deal with the different polarity instances, it might be also required different treatments, such as features or training sets.

These experiment results suggest that using the word and character n-gram combination features could improve the performance of classifiers for detecting deceptive and truthful opinions in positive and negative instances. In other words, providing more comprehensive features for the classifier not only could cover the lack of individual word and character n-gram features but also could boost up the capability of classifiers.

5.2 Experiment 2: Function Words Features

Our purpose to do this experiment is to explore the use of function words combined with other utilized features, i.e., word n-grams, character n-grams, and word and character n-gram combinations, as features in detecting deceptive and truthful opinions. We also expect that by finding out the function words behavior in both deceptive and truthful opinions, it could be useful to distinguish between them. In this experiment, we evaluate the use of function word features on both positive and negative instances of the Chicago hotel dataset [24, 25]. This evaluation is unlike our prior work [33], which exploited the function word features on positive instances only.

Figure 6 shows the obtained classification results with and without combining the function words as features for classifiers. Specifically, the combination of function words with each of 5 char and Uni+5 char as features could improve the classification results in positive instances (Figure 6(a)). In contrast, by combining the function words with other utilized features could not give a meaningful contribution to increasing the classification results, even worse, in negative instances (Figure 6(b)).

To analyze these two distinct results, we investigated two things. First, we took into account the average distribution of function words in each review both deceptive and truthful opinions of positive and negative instances. Second, we observed the most utilized function words in deceptive and truthful opinions.

Results in Table 7 indicate that the average number distribution of function words in each instance probably had no significant differences between deceptive and truthful opinions both in the positive and negative instances. In particular, the average utilized function words in positive

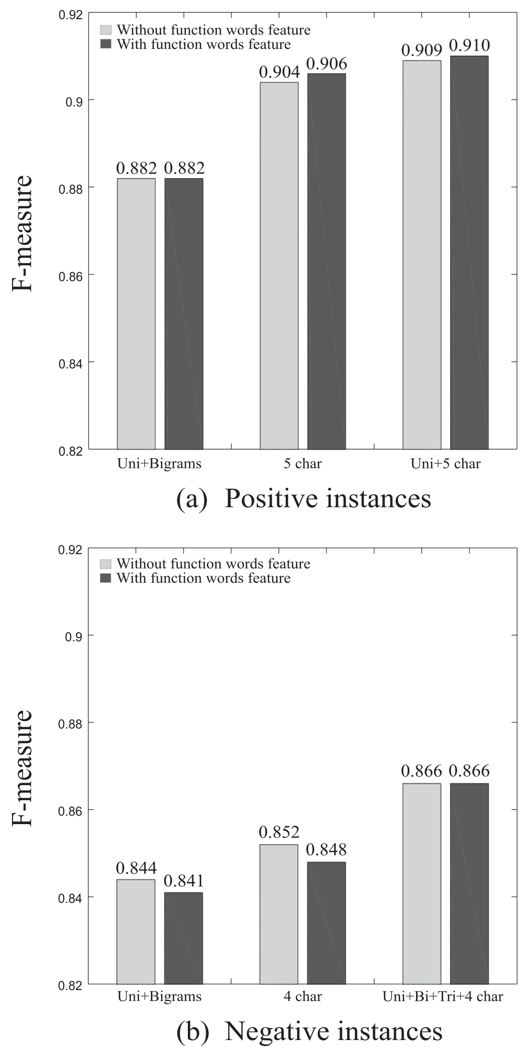

Fig. 6. The obtained classification results with and without combining the function words feature to other features.

truthful opinions were slightly greater than those in positive deceptive ones. Conversely, the negative truthful opinions had a little bit smaller number of function words than the negative deceptive ones. As a result, using the function words as combinations features, i.e., combined with the baseline or our proposed combinations features, could not give a significant impact for classifiers to identify the deceptive and truthful opinions.

Subsequently, Table 8 shows that there might be a small difference between the utilized function words in deceptive opinions and those in truthful opinions for both positive and negative instances. Specifically, the top 10 utilized function words between deceptive and truthful opinions in the positive instances had two differences, i.e., marked in green highlights. Meanwhile, the top 10 utilized function words between deceptive and truthful opinions in the negative instances had one distinction only, viz., highlighted in red. Based on these findings, they might be the cause of using the function words features could help to improve the classification results in the positive

Table 7. The Average Number of Function Words in Each Review of Deceptive and Truthful Opinions for Both Positive and Negative Instances of Chicago Hotel Dataset

The average distribution of function words (FW) in each instance			
Positive		Negative	
Deceptive	Truthful	Deceptive	Truthful
63.6 FW/instance	**65.3 FW/instance**	**103.4 FW/instance**	100 FW/instance

Results in bold indicate the larger average numbers.

Table 8. The Most Utilized Function Words in Deceptive and Truthful Opinions for Both Positive and Negative Instances

Top 10 function words			
Positive		Negative	
Deceptive	Truthful	Deceptive	Truthful
the	the	the	the
and	and	i	to
i	a	to	and
to	was	and	a
a	to	was	i
was	i	a	was
in	**we**	in	in
of	in	of	we
my	of	we	**for**
is	**for**	**my**	of

instances (Figure 6(a)). In contrast, they might be the cause of using the function word features could not affect the classifier performance, even degraded its performance, in negative instances (Figure 6(b)). Nevertheless, note that the classification results of using the word and character n-gram combinations with the function word features seemed stable for detecting deceptive and truthful opinions in the negative instances, even better in the positive instances.

Regrettably, results in Tables 7 and 8 indicate that using the function words as an indicator to identify between deceptive and truthful opinions might not be very effective. In particular, the distribution of function words between deceptive and truthful opinions was probably in the balance for both positive and negative instances (Table 7). To our surprise, the number of utilized function words in negative deceptive opinions was slightly larger than that in the negative truthful ones. This finding might be in contrast with our prior conjecture [33], that is, the reviews are written in a natural way, i.e., the truthful opinions, might have more function words than those of the deceptive opinions. Moreover, the difference of utilized function words in deceptive and truthful opinions might not be significant in both positive and negative instances (Table 8). Specifically, the word "*for*" seemed to be the only function word that might be used more frequently in the truthful opinions than in the deceptive ones for both positive and negative instances.

Although utilizing the function words features could not be contributing to good performance in this second experiment, the use of them might be useful to improve the classification result in detecting deceptive and truthful opinions for positive instances, as well as our prior finding in Reference [33]. Note that, due to the second experimental results, we consider not including the function words as combinations features in our next experiments.

Table 9. Results by Using the Word Syntactic N-grams (sn-word) as Features in Positive and Negative Instances

FEATURES	POSITIVE	NEGATIVE
sn-word	F-measure	F-measure
sn-bi	0.776	0.744
sn-tri	0.607	0.641
Uni+sn-bi	**0.869**	0.840
Uni+sn-bi+sn-tri	0.865	**0.846**

Results in bold indicate the best performance for each polarity.

5.3 Experiment 3: Word Syntactic N-gram Features

The aim of this experiment is to explore the ability of word sn-gram (sn-word) features in detecting deceptive and truthful opinions. Table 9 shows that integrating word sn-bigrams (sn-bi) and word sn-trigrams (sn-tri) with word unigrams (Uni) as features could obtain better classification results than using the sn-bigram and sn-trigram features separately. In particular, the classification results of using Uni+sn-bi features could achieve the best in positive instances, while those of employing Uni+sn-bi+sn-tri features could attain the best in negative ones.

To observe the sn-word features performance, we compared the best-obtained results of using the sn-word with those of our baseline, i.e., the individual of word (word) and character (char) n-grams, and word and character n-gram combination (word+char) features for both positive and negative instances (Figure 7). Overall, the sn-word feature achievement could not exceed the top performance of the word+char features, even for the attainment of using the char features. Moreover, the classification results of employing the sn-word features were worse than those of using the word features in the positive instances (Figure 7(a)). However, Figure 7(b) indicates that the results of using the sn-word features could get better performance than those of utilizing the word features for classifying deceptive and truthful opinions in the negative instances.

The unsatisfactory results of using sn-word features in Figure 7 might be due to the attribute element of the sn-word features. Since constructing the attribute element of the sn-word features is by following syntactic relations, viz., not based on the order of word existences in reviews, the attribute element of the sn-word might not give a particular meaning as an input for classifiers. In other words, the linguistic information for constructing the attribute element of the sn-word could not contribute well in this experiment. As a result, the classifiers probably became more difficult to classify between deceptive and truthful opinions.

Results in Table 10 also indicate that the obtained classification results of using the combination of sn-word with char (snword+char) features could not perform well, as well as those of utilizing the word+char features. However, we believe that the particular characteristic of the sn-word features could be useful in this study. Therefore, we would like to explore further the use of sn-word and sn-word+char features in the next following experiments.

Compared to the excellent classification results of References [37–39], our obtained results of employing sn-word features could not perform well in this experiment. These unexpected outcomes might be due to the difference of characteristics between the utilized dataset in this study and that in References [37–39], that is, novel books. Particularly, the content of opinion reviews might be in unstructured and uncontrolled text forms, e.g., having grammatical mistakes or misspelled words. In contrast, the novel books are mostly well prepared and fully controlled, i.e., through the editing process. Based on this difference, the quality of syntactic trees obtained from

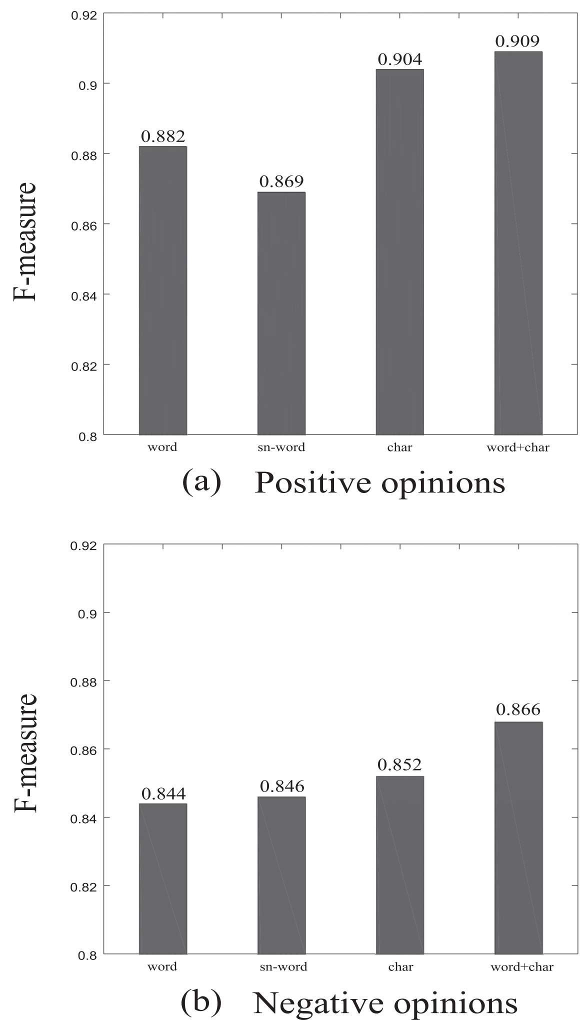

Fig. 7. The comparison among each best-obtained result for word n-grams (word), word sn-grams (sn-word), character n-grams (char), and word and character n-gram combination (word+char) features in detecting positive and negative deceptive opinions.

parsing results might be different. Thus, some ungrammatical forms or error words might degrade the performance of a parser [6, 42]. In fact, the quality degradation of syntactic trees obtained from the parsing results could affect to the poor results of age and gender predictions in an author profiling task [28]. Moreover, a particular number of misspelled words might have a significant influence on the quality of syntactic n-grams. This conjecture is due to the work in Reference [22], where they obtained the lower classification results of using syntactic word n-grams only, i.e., sn-bigrams and sn-trigrams, than those of employing word traditional n-grams. Specifically, Reference [22] identified 39,512 misspelled words in 12,100 essays[15] of their training and development datasets for native language identification task.

[15]The average number of words in each essay was 316.2 [21].

Table 10. The Comparison Between Using the Word+char and Sn-word+char as Features for Classifying Deceptive and Truthful Opinions in Chicago Hotel Dataset

Features	Positive instances	Negative instances
word+char	**0.909**	**0.866**
sn-word+char	0.907	0.862

Results in bold indicate the better-obtained classification results.

Table 11. The Number of Words and Error Words (Corrected) in Chicago Hotel Dataset

Chicago hotel reviews				
	Positive		Negative	
	Deceptive	Truthful	Deceptive	Truthful
Number of words	43746	46744	66964	67951
Number of error words (corrected)	304	711	658	1113

Another reason for our poor-obtained results when using sn-word features in this experiment might be due to a large number of writing styles in our utilized dataset. Since the Chicago hotel dataset was generated by a lot of reviewers [24, 25], the number of writing styles in both deceptive and truthful opinions should be huge. Thus, employing the sn-word features could not work very well in this experiment. This reason corresponds to Reference [38], where the classifier performance of using syntactic n-gram features could be decline when classifying many authors, because there are many writing styles. Nevertheless, it is better to analyze further the performance of sn-words when used as features to detecting deceptive and truthful opinions posted by a small number of reviewers in the future.

5.4 Experiment 4: Original vs. Corrected

To start, we first observe the total number of words and error words, e.g., misspelled or mistyped, which exist in each positive and negative instances for both deceptive and truthful opinions obtained from the Reference [43] application. Results in Table 11 indicate that the truthful opinions tend to have more error words than the deceptive ones. In other words, this finding could indicate that the honest reviews, i.e., truthful opinions, which were generated in a natural manner, normally might contain a lot of error words. This behavior naturally might be in contrast with the untrustworthy ones, that is, deceptive opinions. Due to the purpose of the deceptive opinions, e.g., convincing the readers, writing such opinions is probably done in a careful way. Based on this first observation, it suggested that the error words in reviews might help to identify between deceptive and truthful opinions. Note that the results in Table 11 also confirmed our presumption in Section 5.3 that opinion reviews, viz., our utilized dataset in this work, might have a number of error words.

Table 12 represents the comparison of obtained results for each best achievement of our utilized features in this study when used in the original and corrected datasets. Results in Table 12(a) and (b) indicate that using the word and character n-gram combination features (word+char) could keep consistent to obtain better performance for classifying in the original and corrected datasets, compared to the performance of other features. In particular, the word+char features show robust performance not only in the corrected dataset but also in the real condition, viz., having a lot of noises such as error words in the dataset. Although this unfathomable finding should be investigated further, the obtained results in Table 12 probably might be a valuable argument to

Table 12. The Comparison of the Obtained Results of Our Utilized Features When Applying in the Original and Corrected Datasets

Positive instances		
Features	Original dataset	Corrected dataset
word	0.882	0.880
char	0.904	0.896
sn-word	0.869	*0.872*
word+char	**0.909**	**0.899**
sn-word+char	0.907	0.896

(a)

Negative instances		
Features	Original dataset	Corrected dataset
word	0.844	*0.852*
char	0.852	*0.862*
sn-word	0.846	*0.847*
word+char	**0.866**	***0.877***
sn-word+char	0.862	*0.871*

(b)

Results in bold indicate the best-obtained classification results in the original and corrected datasets. Numbers in italic represent the better-obtained classification results when applying the correction to the dataset.

corroborate our prior conjecture [32]. In other words, the word+char might be reliable features for classifiers to deal with this complicated task.

Furthermore, we note two notable findings from the results in Table 12. First, Table 12(a) indicates that the obtained classification results of using our utilized features in the corrected dataset were worse than those in the original dataset for positive instances. In contrast, Table 12(b) displays that the obtained results by applying the same features to the corrected dataset were better than employing those features to the original dataset for negative instances. According to these two distinct outcomes in Table 12, they might be due to the disparity of the number of error words between positive and negative instances, as shown in Table 11. Specifically, with respect to the number of words for each instance, the number of corrected words in negative instances was slightly larger than that in positive ones. As a result, the corrected words in negative instances might be sufficient to increase the classification results, while those in positive ones might not be able to improve the classification results, and even decreased the classification results. Another rationale is that they might be due to the disappearance of important feature attributes after performing the word correction in the positive instances. In other words, the particular error words might be one of significant clues for classifiers to distinguish between the deceptive and truthful opinions. Consequently, by correcting such error words, it probably removed the most important attributes and predisposed the performance of classifiers. For these reasons, applying the word correction to the dataset could be unnecessary to address this task.

However, we also note an interesting finding in Table 12(a) and (b). Using only sn-words as features for classifying in the corrected dataset consistently resulted in better classification results than in the original dataset both on positive and negative instances. In particular, this performance of the sn-word features was in contrast with that of other features in positive corrected instances,

Table 13. The Obtained Classification Results When Performing Cross-classifications

	Cross-classifications			
Features	Cross-hotel	Cross-domain	Positive cross-type	Negative cross-type
word	0.764	0.669	0.769	0.682
char	**0.773**	0.683	0.791	0.689
sn-word	0.750	0.679	0.785	0.683
word+char	**0.773**	0.690	**0.793**	0.689
sn-word+char	0.771	**0.696**	0.791	**0.690**

Results in bold indicate the best obtained classification results for each classification test.

viz., the performance of other features in the positive corrected instances was worse than that in the positive original ones. Based on this notable finding, performing the word correction to the dataset might give a beneficial value to enhance the ability of the sn-word features, i.e., word syntactic n-grams, in this task. Nonetheless, it is better to study further this finding in the future.

5.5 Experiment 5: Cross-Classifications

Our main consideration to do this experiment is to examine further the performance of our proposed features, namely, sn-words, word+char, and sn-word+char, when employed in other scopes, i.e., other hotels, other domains, and the real-world reviews. Results in Table 13 indicate that overall the performance of word+char and sn-word+char features could consistently outperform the baseline features, i.e., word and char, when classifying in cross-classification tests. In particular, using the word+char features could obtain the best classification results in the cross-hotel and positive cross-type classifications, whereas utilizing the sn-word+char features could gain the highest achievement for the cross-domain and negative cross-type classifications.

Based on the results in Table 13, we note an interesting finding. The finding is that the performance of utilizing the sn-word+char features for classifying in the cross-domain and negative cross-type classifications might not only be the best but also could surpass that of employing the word+char features. This outcome was surprising, since the performance of the sn-word+char features in our previous experiments, i.e., using the Chicago hotel dataset only, might be inferior to that of the word+char features. In addition, the obtained results of the syntactic word n-grams (sn-word) were better than those of the traditional word n-grams (word), viz., the baseline features, when utilized for classifying in cross-domain and both positive and negative cross-type classifications. According to these findings, they might be due to the utilized attribute element of the sn-word features. Since constructing the attribute element of the sn-word features is relatively different with those of the word features, the use of the sn-word features as an input for classifiers could give a more benefit when used in other fields, i.e., not about hotel reviews only. In other words, the attribute element of the sn-word features might be more flexible and applicable when applied in various circumstances. Indeed, this valuable finding conforms to the purpose of the syntactic n-grams approach that is to reduce the ineffectual of features attributes when generated by the traditional n-grams [37–39]. Nevertheless, since we used continuous sn-grams [37–39] only in this study, it would be valuable to explore another type of sn-grams, namely, non-continuous[16] sn-grams [29, 34–36], in the future.

To summarize, these last experimental results suggest the usefulness of our proposed combinations features when utilized as an input for classifiers to detect the deceptive and truthful opinions not only in a specific field, i.e., Chicago hotel reviews, but also in other objects, viz., other hotels,

[16]Syntactic n-grams with bifurcations.

other domains, and the real-world reviews. The results also confirm our belief that performing combinations between word and character n-grams features, i.e., word+char and sn-word+char, might cover and fulfill each lack of them. Therefore, we believe that our proposed combinations features might be a robust input for classifiers in detecting deceptive and truthful opinions.

6 CONCLUSION

This study has investigated the use of word and character n-gram combinations, function words, and word sn-grams as features in detecting deceptive and truthful opinions. It then has applied word correction to our employed dataset. Our obtained results in the first to fourth experiments indicated that using the word and character n-gram combination features could perform better compared to those of employing other features. Unfortunately, the results also seemed to suggest that it might be unimportant to apply the word correction in this task. However, we noted that the number of error words in the deceptive opinions appeared smaller than that in the truthful ones. To check robustness of our utilized features, we then carried out cross-classification tests. Our last experimental results would seem to suggest that the word and character n-gram combination could be robust features in detecting deceptive dan truthful opinions. Interestingly, we also found that using the word sn-grams as combination features could exhibit good performance in this study.

In the future, we believe that it must be valuable to inspect further our work to other deceptive texts, e.g., controversial opinions [16, 27] and news [1]. Future work will also consider using character n-grams in tokens[17] features [2, 3] as combinations with other features, e.g., word n-grams and word sn-grams. This consideration is motivated by References [2, 3], which utilizing the character n-grams in tokens as combination features could obtain good classification results in their comprehensive work. In addition, the number of attributes of character n-grams in tokens is considerably smaller than that of the traditional character n-grams [2, 3]. However, we think it should be important to study deceiver writing styles, as well as behavior features [23], to improve the classification results in this deception detection task.

ACKNOWLEDGMENTS

The authors acknowledge the anonymous reviewers for their valuable comments that significantly improved this work.

REFERENCES

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Secur. Priv.* 1, 1 (2018), e9. DOI: https://doi.org/10.1002/spy2.9

[2] Leticia C. Cagnina and Paolo Rosso. 2015. Classification of deceptive opinions using a low dimensionality representation. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA@EMNLP'15)*. Association for Computational Linguistics, Stroudsburg, PA, 58–66. DOI: https://doi.org/10.18653/v1/W15-2909

[3] Leticia C. Cagnina and Paolo Rosso. 2017. Detecting deceptive opinions: Intra and cross-domain classification using an efficient representation. *Int. J. Uncertain. Fuzz. Knowl.-Based Syst.* 25, Supplement-2 (2017), 151–174. DOI: https://doi.org/10.1142/S0218488517400165

[4] Michael Crawford, Taghi M. Khoshgoftaar, Joseph D. Prusa, Aaron N. Richter, and Hamzah Al Najada. 2015. Survey of review spam detection using machine learning techniques. *J. Big Data* 2, 1 (Oct. 2015), 2–23. DOI: https://doi.org/10.1186/s40537-015-0029-9

[5] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*. ELRA, Paris, 449–454.

[6] Jennifer Foster. 2010. "Cba to check the spelling" investigating parser performance on discussion forum posts. In *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the*

[17] Set of characters that is segregated by space(s) [2, 3].

Association for Computational Linguistics (HLT '10). Association for Computational Linguistics, Stroudsburg, PA, 381–384.

[7] Donato Hernández Fusilier, Rafael Guzmán-Cabrera, Manuel Montes-y-Gómez, and Paolo Rosso. 2013. Using PU-learning to detect deceptive opinion spam. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA@NAACL-HLT 2013)*. Association for Computational Linguistics, Stroudsburg, PA, 38–45.

[8] Donato Hernández Fusilier, Manuel Montes-y-Gómez, Paolo Rosso, and Rafael Guzmán Cabrera. 2015. Detecting positive and negative deceptive opinions using PU-learning. *Inf. Process. Manage.* 51, 4 (Jul. 2015), 433–443. DOI: https://doi.org/10.1016/j.ipm.2014.11.001

[9] Donato Hernández Fusilier, Manuel Montes y Gómez, Paolo Rosso, and Rafael Guzmán Cabrera. 2015. Detection of opinion spam with character n-grams. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'15)*. Springer International Publishing, 285–294. DOI: https://doi.org/10.1007/978-3-319-18117-2_21

[10] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. DOI: https://doi.org/10.1145/1656274.1656278

[11] Christopher G. Harris. 2012. Detecting deceptive opinion spam using human computation. In *Proceedings of the 4th AAAI Human Computation Workshop (HCOMP'12)*. AAAI, 87–93.

[12] S. A. Jadhav, D. V. L. N. Somayajulu, S. N. Bhattu, R. B. V. Subramanyam, and P. Suresh. 2013. Topic dependent cross-word spelling corrections for web sentiment analysis. In *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI'13)*. IEEE Computer Society, 1093–1096. DOI: https://doi.org/10.1109/ICACCI.2013.6637329

[13] Nitin Jindal and Bing Liu. 2007. Analyzing and detecting review spam. In *Proceedings of the 2007 7th IEEE International Conference on Data Mining (ICDM'07)*. IEEE Computer Society, 547–552. DOI: https://doi.org/10.1109/ICDM.2007.68

[14] Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM, New York, NY, 1189–1190. DOI: https://doi.org/10.1145/1242572.1242759

[15] Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM'08)*. ACM, New York, NY, 219–230. DOI: https://doi.org/10.1145/1341531.1341560

[16] Javier Sánchez Junquera, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez, and Paolo Rosso. 2018. Character n-grams for detecting deceptive controversial opinions. In *Proceedings of the 9th International Conference of the CLEF Association (CLEF'18)*. Springer, 135–140. DOI: https://doi.org/10.1007/978-3-319-98932-7_13

[17] Mike Kestemont. 2014. Function words in authorship attribution. From black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL'14)*. Association for Computational Linguistics, Gothenburg, Sweden, 59–66. DOI: https://doi.org/10.3115/v1/W14-0908

[18] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.* 60, 1 (Jan. 2009), 9–26. DOI: https://doi.org/10.1002/asi.v60:1

[19] Jiwei Li, Myle Ott, and Claire Cardie. 2013. Identifying manipulated offerings on review portals. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. Association for Computational Linguistics, Stroudsburg, PA, 1933–1942. http://aclweb.org/anthology/D/D13/D13-1199.pdf

[20] Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. 2014. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 1566–1576. DOI: https://doi.org/10.3115/v1/P14-1147

[21] Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A report on the 2017 native language identification shared task. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Copenhagen, Denmark, 62–75. DOI: https://doi.org/10.18653/v1/W17-5007

[22] Ilia Markov, Lingzhen Chen, Carlo Strapparava, and Grigori Sidorov. 2017. CIC-FBK approach to native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (BEA'17)*. Association for Computational Linguistics, Stroudsburg, PA, 374–381. DOI: https://doi.org/10.18653/v1/W17-5042

[23] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. What yelp fake review filter might be doing? In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM'13)*. 10.

[24] Myle Ott, Claire Cardie, and Jeffrey T. Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'13)*. Association for Computational Linguistics, Stroudsburg, PA, 497–501.

[25] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies—Volume 1 (HLT '11). Association for Computational Linguistics, Stroudsburg, PA, 309–319. http://dl.acm.org/citation.cfm?id=2002472.2002512

[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830. http://dl.acm.org/citation.cfm?id=1953048.2078195

[27] Veronica Pérez-Rosas and Rada Mihalcea. 2014. Cross-cultural deception detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, 440–445.

[28] Juan Pablo Posadas-Durán, Helena Gómez-Adorno, Ilia Markov, Grigori Sidorov, Ildar Z. Batyrshin, Alexander F. Gelbukh, and Obdulia Pichardo-Lagunas. 2015. Syntactic n-grams as features for the author profiling task: Notebook for PAN at CLEF 2015. In *Proceedings of the Working Notes of CLEF 2015—Conference and Labs of the Evaluation Forum.*

[29] Juan Pablo Posadas-Duran, Grigori Sidorov, and Ildar Batyrshin. 2014. Complete syntactic n-grams as style markers for authorship attribution. In *Human-Inspired Computing and Its Applications*. Springer International Publishing, Cham, 9–17.

[30] Juan Pablo Posadas-Durán, Grigori Sidorov, Helena Gómez-Adorno, Ildar Z. Batyrshin, Elibeth Mirasol-Meléndez, Gabriela osadas Durán, and Liliana Chanona-Hernández. 2017. Algorithm for extraction of subtrees of a sentence dependency parse tree. *Acta Polytech. Hung.* 14, 3 (2017), 79–98. DOI: https://doi.org/10.12700/APH.14.3.2017.3.5

[31] Stacey Rudolph. 2015. The impact of online reviews on customers' buying decisions. Retrieved March 1, 2018 from https://www.business2community.com/infographics/impact-online-reviews-customers-buying-decisions-infographic-01280945.

[32] Al Hafiz Akbar Maulana Siagian and Masayoshi Aritsugi. 2017. Combining word and character n-grams for detecting deceptive opinions. In *Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC'17)*. IEEE Computer Society, 828–833. DOI: https://doi.org/10.1109/COMPSAC.2017.90

[33] Al Hafiz Akbar Maulana Siagian and Masayoshi Aritsugi. 2018. Exploiting function words feature in classifying deceptive and truthful reviews. In *Proceedings of the 2018 13th International Conference on Digital Information Management (ICDIM'18)*. IEEE Computer Society, 51–56. https://doi.org/10.1109/ICDIM.2018.8846971

[34] Grigori Sidorov. 2013. Non-continuous syntactic n-grams. *Polibits* 48 (2013), 69–78.

[35] Grigori Sidorov. 2013. *Non-linear Construction of N-grams in Computational Linguistics: Syntactic, Filtered, and Generalized N-grams*. Sociedad Mexicana de Inteligencia Artificial.

[36] Grigori Sidorov. 2019. *Syntactic N-grams in Computational Linguistics*. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-14771-6

[37] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2013. Syntactic dependency-based n-grams as classification features. In *Proceedings of the 11th Mexican International Conference on Advances in Computational Intelligence—Volume Part II (MICAI'12)*. Springer-Verlag, Berlin, 1–11. DOI: https://doi.org/10.1007/978-3-642-37798-3_1

[38] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2013. Syntactic dependency-based n-grams: More evidence of usefulness in classification. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'13)*. Springer-Verlag, Berlin, 13–24. DOI: https://doi.org/10.1007/978-3-642-37247-6_2

[39] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2014. Syntactic n-grams as machine learning features for natural language processing. *Exp. Syst. Appl.* 41, 3 (Feb. 2014), 853–860. DOI: https://doi.org/10.1016/j.eswa.2013.08.015

[40] Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol.* 60, 3 (Mar. 2009), 538–556. DOI: https://doi.org/10.1002/asi.v60:3

[41] Alexandra Topping. 2010. Historian Orlando Figes agrees to pay damages for fake reviews. Retrieved March 1, 2018 from https://www.theguardian.com/books/2010/jul/16/orlando-figes-fake-amazon-reviews.

[42] Joachim Wagner and Jennifer Foster. 2009. The effect of correcting grammatical errors on parse probabilities. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*. Association for Computational Linguistics, Stroudsburg, PA, 176–179. http://dl.acm.org/citation.cfm?id=1697236.1697270

[43] Zar Zar Wint, Theo Ducros, and Masayoshi Aritsugi. 2017. Spell corrector to social media datasets in message filtering systems. In *Proceedings of the 2017 12th International Conference on Digital Information Management (ICDIM'17)*. IEEE Computer Society, 209–215. DOI: https://doi.org/10.1109/ICDIM.2017.8244677

Received April 2018; revised June 2019; accepted July 2019