Identifying Deceptive Reviews: Feature Exploration, Model Transferability and Classification Attack

Marianela García Lozano*† and Johan Fernquist*
*FOI Swedish Defence Research Agency, SE-164 90 Stockholm, Sweden
†KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
Email: {garcia, johan.fernquist}@foi.se

***Abstract*—The temptation to influence and sway public opinion most certainly increases with the growth of open online forums where anyone anonymously can express their views and opinions. Since online review sites are a popular venue for opinion influencing attacks, there is a need to automatically identify deceptive posts.**

The main focus of this work is on automatic identification of deceptive reviews, both positive and negative biased. With this objective, we build a deceptive review SVM based classification model and explore the performance impact of using different feature types (TF-IDF, word2vec, PCFG). Moreover, we study the transferability of trained classification models applied to review data sets of other types of products, and, the classifier robustness, i.e., the accuracy impact, against attacks by stylometry obfuscation trough machine translation.

Our findings show that i) we achieve an accuracy of over 90% using different feature types, ii) the trained classification models do not perform well when applied on other data sets containing reviews of different products, and iii) machine translation only slightly impacts the results and can not be used as a viable attack method.

***Index Terms*—Deceptive; fake; classification; SVM; Word2vec; PCFG.**

I. Introduction

The growth of online forums, online booking agencies, and online shops have made them popular venues for advertisers, marketers, and such trying to influence people both into believing that either their products or services are the best one offered, or that competitors' are worse. The great potential of online market evaluations was pointed out already in 1999 [1].

In 2012, it was believed that by 2014, 10% to 15% of all reviews on social media would be fake and payed for by companies [2]. In late 2018, it was reported that 33% of the participants of a survey regarding fake news reported that they had spotted lots of fake reviews online [3]. TripAdvisor reported in their *Review Transparency Report 2019* that 2.1% of the submitted reviews on the platform were fake reviews and that 91% of the fake reviews were positive biased [4].

Forum posts can be contradictory, false, ambiguous and biased for a number of reasons. This work does not explore those reasons but focuses on automatically identifying the deceptive reviews. A deceptive review is sometimes also called a fake review, i.e., a review written by someone who conveys a positive or negative message on their own or someone else's behalf for someone's profit.

The research questions we address in this work are:

- What impact do different features have on classification of deceptive reviews using a linear support vector machine (SVM) based classification model performance?
- How well do trained models transfer in terms of classification accuracy between different product review data sets?
- How does machine translation influence the stylometric, i.e., linguistic, patterns of deception and truthfulness, and what consequences does that have on the classification accuracy?

The remainder of this paper is organized as follows: Section II describes related work. The design and implementation choices are motivated in Section III. The experiment setups are described in Section IV. In Section V the results and evaluation is presented. An in depth discussion of the results is presented in Section VI and concluding remarks wrap up the paper in Section VII.

II. Related Work

Many of the earlier deceptive review text classification research efforts focused on the implications of the existence of deceptive reviews, discovering and defining the similarities and differences between deceptive and truthful reviews [5], [6]. Recent work has focused more on automatic review classification [7]–[10]. Most of those researchers have used the same data set, which consists of reviews of the 20 most popular hotels in the Chicago area collected from TripAdvisor combined with artificially created deceptive reviews from Amazon Mechanical Turk (AMT). In [7] linguistic inquiry and word count (LIWC) is used in combination with bi-grams on an SVM. A finding from that work is that deceptive reviews are more likely to hold an excessive amount of positive or negative words [7]. In a later effort the same researchers expanded the data set and used standard n-gram based SVM [9]. The original AMT data set was also used in [8]. The classification accuracy result is improved by using Probabilistic Context Free Grammar (PCFG) as input for the SVM algorithm. With the help of their linguistics students [10] created a new data set in Dutch. They used a linear SVM with bigrams as features to detect deceptive reviews.

In [11] the authors focused on a slightly different facet of the problem and proposed a method to detect fake reviewers using unsupervised Bayesian inference framework.

EISIC 2019, November 26-27, 2019, Oulu, Finland
978-1-7281-6735-0/19/$31.00 © 2019 IEEE

SVM in combination with word2vec has not been done in large scale, and to the best of our knowledge never with the purpose to detect deception. A related work with the purpose to cluster newsgroup posts according to category was done by [12]. They employed both term frequency-inverse document frequency (TF-IDF) and word2vec, both alone and in combination with an SVM. They also trained both with and without stop words and saw a notable difference in accuracy.

On several occasions where the data has consisted of text documents and the features are of stylometric nature SVM has outperformed other classification algorithms such as K-Nearest Neighbors and Naive Bayes, [13], [14]. The J48 algorithm, which generates a decision tree, and SVM have outperformed each other on different kinds of data [15]. For binary outcomes, the algorithms SVM, Random Forest and Adaptive Boosting are known to be ideal for classification of text documents [16].

In other types of related work researchers have focused on deceptive texts, e.g., spam, where the text authors have tried to attack fake text classifiers by hiding their own writing style [15]. Examples of features used for author deanonymization are average number of words per sentence, number of characters per words, and frequency of long words and percentage of letters and digits. Content specific features have also been used where the topic of a word is taken into account. In a related work, [17] used machine translation to try and anonymize texts by hiding the authors' stylometric patterns. The group used *Google Translate* and *Bing* to do one- and two-step translations to either German, Japanese or both, and then back to English. Their results gave an accuracy drop between 15% and 35%, and in some cases with a loss of the text's intention.

III. Design and Implementation

To automatically differentiate between truthful and deceptive reviews we made some design and implementation choices. This Section describes and motivates these choices.

A. Data Sets

Two different data sets were used in this work. One of the data sets is the same that was used in [9]. The data set, called AMT, consists of 800 truthful and 800 deceptive hotel reviews, all written in English. Half of the reviews are positive, i.e., corresponding to 5-stars ratings, and the other half negative, i.e., corresponding to 1- or 2-stars ratings. The positive truthful reviews were collected from TripAdvisor and the negative truthful reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp. All deceptive reviews (both positive and negative) were created using Amazon's Mechanical Turk where people got paid to write them. The reviews' distribution is shown in Table I.

TABLE I
AMT: hotel review data set class composition.

	Deceptive	Truthful
Positive	400	400
Negative	400	400

The other data set we used is the so called CLiPS stylometry investigation corpus (CSI) [10]. It is a Dutch written corpus holding both essays and reviews. The review part of the data set holds 1298 reviews, both deceptive and truthful, and positive and negative. The review distribution is shown in Table II. All reviews were written by students taking Dutch proficiency courses at the university of Antwerpen and the topics of the reviews are musicians, food chains, books, smartphones and movies.

TABLE II
CSI: musicians, food chains, books, smartphones and movies review data set class composition

	Deceptive	Truthful
Positive	319	323
Negative	330	326

Both the data sets have fairly balanced truthful, deceptive, positive and negative classes. For ease of use, purposes we translated the CSI texts from Dutch to English. One drawback of using translated reviews for the classification tasks is that sentiment patterns may not be translated accordingly.

A subscript index, see Table III, is used throughout the paper corresponding to the data set class, how the data was translated or which part of the data set was used for testing. For example, $\text{AMT}_{\text{eng}\rightarrow\text{rus}\rightarrow\text{eng}}$ means that we have translated the AMT data set from English to Russian and then back to English again.

TABLE III
Data set subscript definition

Subscript	Definition
$\text{Data set}_{\text{language1}\rightarrow\text{language2}}$	The data set has been translated from language 1 to language 2
Data set_{+}	Only the positive reviews (both truthful and deceptive)
Data set_{-}	Only the negative reviews (both truthful and deceptive)
$\text{Data set}_{\text{TO+}}$	Only positive reviews of the data set used for tests
$\text{Data set}_{\text{TO-}}$	Only negative reviews of the data set used for tests

B. Classification Algorithm

On several occasions a linear kernel Support Vector Machine (SVM) has been shown to be a good algorithm for text classification [14], [16]. In all to us known other work using the same data sets that we employ SVM, has outperformed the other algorithms [7]–[9], making SVM a natural choice for our task.

To find the optimal set of hyperparameters it is common to apply grid search. Hence, grid search with cross validation (CV), i.e., k-fold CV is applied for the classification model training.

C. Preprocessing

To ensure that the input data will have minimum noise, i.e., unnecessary information, which can impair the outcome of the classification some preprocessing of the data is desirable.

Stop words are commonly occurring words. These common words are filtered out because they are considered to have a small impact during the training and classification. A drawback may be that patterns containing stop words and representing deception can be systematically removed as part of this preprocessing stage. Due to this potential drawback experiments with and without stop words will be done.

There is no universal list of stop words since the words considered as stop words can be chosen differently. Within stop words, it is common to chose words from the word classes pronouns and prepositions. Other common stop words in the English language are words such as ”a, an, and, but, or” and common verbs. The stop words used in this work are the ones provided by the NLTK package [18]. We also used the package’s stemming functions since others have used it in conjunction with SVM and have seen some improvement in results [19]. Stemming is the method for grouping together different forms of a word such as ”catching”, ”catches” and ”catch” to be treated as one single item.

To summarize we have the following preprocessing combinations:

- P_1 = unstemmed without stop words
- P_2 = unstemmed with stop words
- P_3 = stemmed without stop words
- P_4 = stemmed with stop words

D. Features

SVM uses a vector space model representation of each document, i.e., a feature vector f such as $[f_1, f_2, \ldots, f_n]$ where f_i is the value of feature i and n the number of features. Based on the features and results other have achieved with them we have chosen three main types.

The first feature type is Term Frequency-Inverse Document Frequency also known as TF-IDF [20]. The TF-IDF of term t in document d yields

$$\mathrm{TF-IDF}_{t,d} = \log f_{t,d} \cdot \log \frac{N}{n_t}$$

where $f_{t,d}$ is the frequency of t in d, N is the total number of documents and n_t is the number of documents that t occurs in. In our work a document d is a review, t are the used words and N is the total number of reviews in a data set.

The second feature type is the output vectors of the word embedding tool word2vec. Word2vec is a neural network which is used to calculate the semantic proximity of words [21]. Each word in the corpus receives a multidimensional space point where words used in similar contexts are clustered together. For each review, the vectors for each of the words appearing in the review are summarized and used as the feature vector such as

$$f(d_i) = \sum_{t=1}^{T_i} \mathrm{word2vec}(t)$$

where T_i are all the terms in document i and word2vec(t) the output vector from word2vec of term t. Each review obtains a multidimensional vector which is used for training the SVM model. To use word2vec we chose the free Python library Gensim [22]. The word2vec model used in this work was trained by us on a sample of English Wikipedia.

The third feature type used is the Probabilistic Context Free Grammar (PCFG) method [23]. The difference between PCFG and the other feature types is that while the other focus on the words themselves and their relations, PCFG focuses on the structural buildup of the sentences.

The Berkeley parser is used to parse sentences [24]. It has a grammar which is trained on texts from the Wall Street Journal. For a given sentence, the parser builds a tree of the grammatical rules with the highest probabilities corresponding to the sentence which is then returned. The feature vectors are generated by encoding the rule lists as TF-IDF values where every rule is treated as a term. This method of using TF-IDF on the rule lists is, e.g., used in [8]. Every rule consist of a left hand side (LHS), an arrow $\rightarrow$, and a right hand side (RHS). The LHS is always a non-terminal whereas the RHS is either one or several terminals or non terminals. For PCFG, a terminal is a word and a non-terminal is a string corresponding to a single or combination of word classes such as noun phrase (NP) or verb phrase (VB). In some cases we are including the grandparent node which is denoted with a following $\wedge$. The grandparent node is always a non-terminal.

To summarize we have used four different types of PCFG rules:

- R_1 = Unlexical rules, i.e., all rules except those where the RHS is a terminal.
  E.g., $NP \rightarrow Noun$
- R_2 = Lexical rules, i.e., all rules including those where the RHS is a terminal.
  E.g., $Det \rightarrow$ 'a'
- R_3 = Unlexical rules with grandparent node, i.e., all rules except those where the RHS is a terminal with the grandparent node.
  E.g., $S \wedge NP \rightarrow Noun$
- R_4 = Lexical rules with grandparent node, i.e., all rules including those where the RHS is a terminal with the grandparent node.
  E.g., $NP \wedge Det \rightarrow$ 'a'

Finally we also tried concatenating feature vectors from the three main feature types such as:

$$f = [f_1^1, f_2^1, \ldots, f_n^1, f_1^2, f_2^2, \ldots, f_m^2]$$

where f^1 and f^2 are two feature vectors of length m and n respectively, which were merged to a new feature vector f.

The Python scikit-learn library functions are used for SVM, feature selection and TF-IDF.

E. Feature Selection

Classification can perform poorly when there are too many superfluous features. By trimming the number of features and removing irrelevant features, In [25] the authors got good results in their classification work using the χ^2-test [26]. Hence, in line with their work we use a feature reduction method where the K number of features with the highest value is selected using the χ^2-test. The χ^2-test measures feature dependency of class and value difference from an expected value based on the null hypothesis that there should be no correlation between features and class. The tests are performed in an incremental manner where the number of features are iteratively increased.

IV. Experiment Setup

To explore the research questions stated in Section I three main types of experiment setups were devised and implemented.

A. Feature Exploration

In order to use the exact same data setup as in [9], AMT is divided into two sets, $\text{AMT}_{\text{TO+}}$ and $\text{AMT}_{\text{TO-}}$ where the index indicates which part of the data that is used to test on. In their work, both the AMT_{+} and AMT_{-} data sets are divided into five subsets where four of the subsets from each data set are used for training. In the $\text{AMT}_{\text{TO+}}$ case, the last fifth part of AMT_{-} is completely held out and the last fifth part of AMT_{+} is used for testing, for that part of the 5-fold cross validation. This means that we train on 1280 reviews and test on 160 reviews for the combined cases.

For TF-IDF and PCFG, we use feature selection and test different number of features. For word2vec, we test for different dimensions of the word vectors. To use PCFG, a grammar file for the used language is required. Due to the lack of an available grammar file for Dutch at the time of experimenting we chose to use PCFG on $\text{CSI}_{\text{dut}\rightarrow\text{eng}}$ instead of CSI_{dut}.

In summary we experiment with the following combinations of data sets and settings, i.e., preprocessing and feature types:

- TF-IDF both with and without stemming and with and without stop words included (CSI, AMT_{+}, AMT_{-}, $\text{AMT}_{\text{TO+}}$, $\text{AMT}_{\text{TO-}}$)
- Word2vec both with and without stemming and with and without stop words included (CSI, AMT_{+}, AMT_{-}, $\text{AMT}_{\text{TO+}}$, $\text{AMT}_{\text{TO-}}$)
- PCFG with stop words and without stemming both with and without lexicalized nodes and grandparent nodes ($\text{CSI}_{\text{dut}\rightarrow\text{eng}}$, AMT_{+}, AMT_{-}, $\text{AMT}_{\text{TO+}}$, $\text{AMT}_{\text{TO-}}$)
- A combination of the feature-vectors generated from the methods giving the best accuracies ($\text{CSI}_{\text{dut}\rightarrow\text{eng}}$, AMT_{+}, AMT_{-}, $\text{AMT}_{\text{TO+}}$, $\text{AMT}_{\text{TO-}}$ and AMT in combination with $\text{CSI}_{\text{dut}\rightarrow\text{eng}}$)

B. Classification Model Transferability

To test how well the classification models perform on other data sets than the one trained on, we performed two tests. We took the best feature combination from the experiments described in Section IV-A and did the following orthogonal tests:

- trained on AMT and tested on $\text{CSI}_{\text{dut}\rightarrow\text{eng}}$
- trained on $\text{CSI}_{\text{dut}\rightarrow\text{eng}}$ and tested on AMT

C. Translation Impact

To explore the possibilities of classification attacks through machine translation data set obfuscation we experiment with translation to see the impact it may have on the classifier performance. The best classifier from Section IV-A is chosen and for both of the data sets only the test data is translated. Classifications of AMT and $\text{CSI}_{\text{dut}\rightarrow\text{eng}}$ are used as the base line. The languages chosen for the translation experiments are: English, Russian and Swedish. Swedish is chosen because of its structural similarities to English and Dutch, and Russian because of its differences to the other languages. Swedish, English and Dutch are Germanic languages, and Russian belongs to the Slavic language group. For each data set, the following translations are made: i) $Eng \rightarrow Swe \rightarrow Eng$, ii) $Eng \rightarrow Rus \rightarrow Eng$, and iii) $Eng \rightarrow Swe \rightarrow Rus \rightarrow Eng$.

The hypothesis is that the more times the data is translated, the more is its stylometric footprint hidden which should result in an accuracy decrease. Another hypothesis is that due to the similarities between English and Swedish, the Swedish translation may perform better than the Russian one. This can occur at the expense of the quality of the texts regarding sentence structure and comprehensibility.

V. Results and Evaluation

The classification performance evaluation is done by measuring both accuracy and F-score. In [27] all experiment results can be found. As they are too many to include only a carefully selected amount of results are presented in this section.

A. Feature Exploration

In Figure 1 the accuracy plot for the AMT_{+} data set can be seen. The number of features for each feature type varies. The accuracy pattern between the different feature types that is seen in the figure was present in all data set experiments. For data set result comparison reasons we present the AMT_{+} and CSI data sets tables which contain the best accuracy and F-score for each feature type and setting, see Table IV and Table V.

TABLE IV
Best AMT_{+} accuracy (A) and F-score (F) results

Feat. type	Settings	A	F
TF-IDF	P_1	89.88	90.42
	P_2	**90.63**	**91.85**
	P_3	90.00	89.75
	P_4	90.25	91.25
word2vec	P_1	84.12	83.04
	P_2	83.75	84.45
	P_3	**85.62**	**85.18**
	P_4	**85.62**	83.29
PCFG	R_1	75.75	78.04
	R_2	90.62	**88.11**
	R_3	76.75	80.02
	R_4	**91.13**	86.85
TF-IDF + PCFG	P_2 + R_2	**91.50**	**93.87**

From the performance of the different feature types, we decided to combine the top two feature types, i.e., TF-IDF and PCFG, with the ambition to further boost the results.

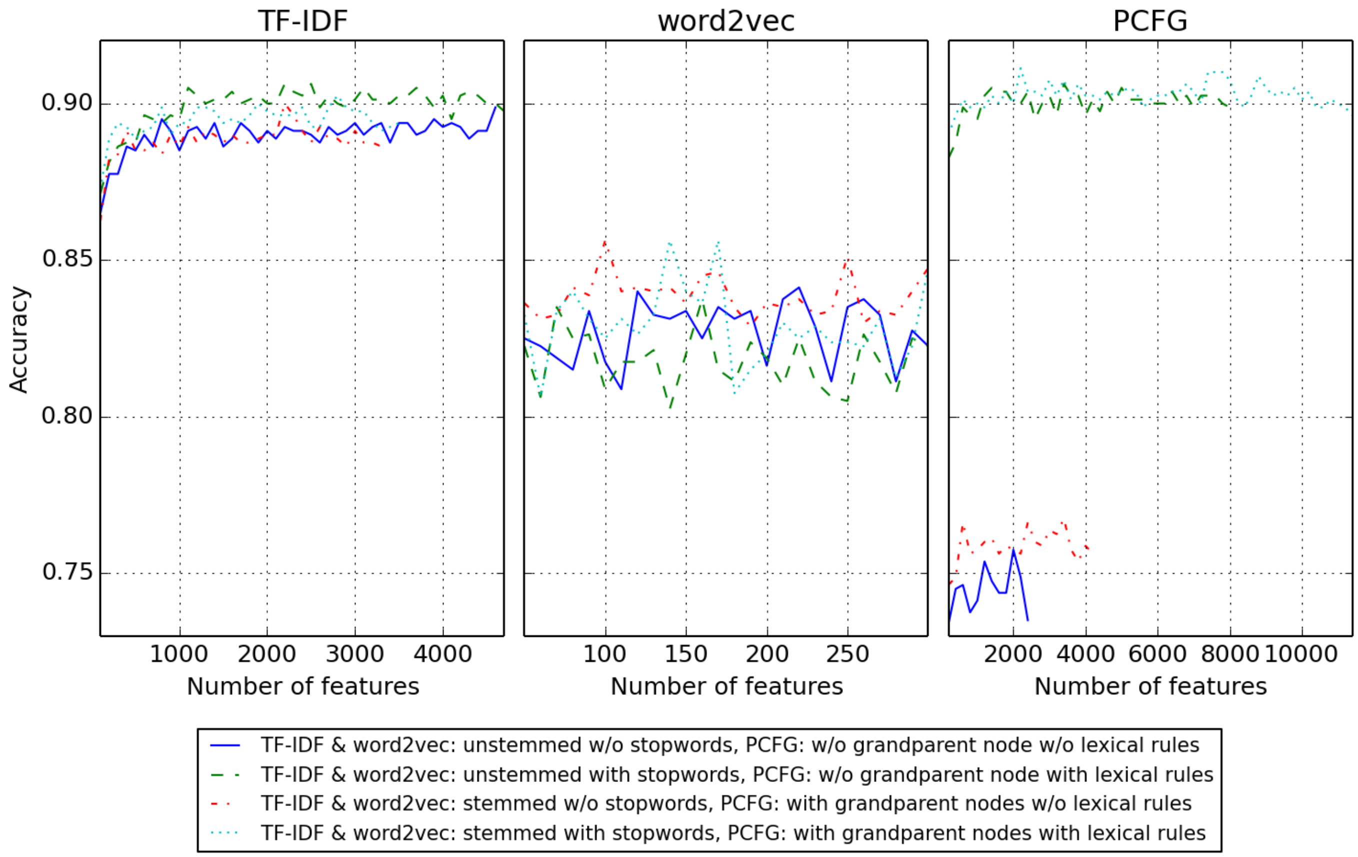

Fig. 1. Accuracy for the different models on AMT_+

TABLE V
BEST CSI ACCURACY (A) AND F-SCORE (F) RESULTS

Feat. type	Settings	A	F
TF-IDF	P_1	82.73	84.86
	P_2	**83.65**	83.83
	P_3	83.16	**85.68**
	P_4	82.73	83.58
word2vec	P_1	75.12	**80.61**
	P_2	71.04	73.71
	P_3	**75.26**	76.13
	P_4	71.88	73.73
PCFG	R_1	62.00	59.83
	R_2	**83.28**	83.64
	R_3	62.42	60.38
	R_4	82.68	**86.10**
TF-IDF + PCFG	P_2 + R_2	**84.67**	**81.73**

A compilation of all data sets' performance measurements obtained with this new classifier can be seen in Table VI. We also combined the two data sets, i.e., AMT and $CSI_{dut \rightarrow eng}$, to obtain a larger data set, which can be seen at the bottom of Table VI.

B. Classification Model Transferability

In Table VII the model transferability experiment results can be seen. Accuracy and F-score are drastically lower for these experiments.

TABLE VI
BEST CLASSIFIER COMBINATION PERFORMANCE MEASUREMENTS FOR ALL DATA SET VARIATION EXPERIMENTS

Feat. type	Settings	Data set	A	F
TF-IDF + PCFG	P_2 + R_2	AMT_+	91.50	93.87
		AMT_-	90.13	88.17
		AMT_{TO+}	91.00	93.08
		AMT_{TO-}	89.12	87.74
		$CSI_{dut \rightarrow eng}$	84.67	81.73
		Combined	85.51	86.81

TABLE VII
ACCURACY (A) AND F-SCORE (F) PERFORMANCE MEASURE OF TRANSFER EXPERIMENTS

Data set	A	F
Trained on AMT tested on $CSI_{dut \rightarrow eng}$	53.78	30.23
Trained on $CSI_{dut \rightarrow eng}$ tested on AMT	53.44	64.37

C. Translation Impact

To discern the quality of the review translations we sampled some of them and compared the originals to the translated ones. In the following three-way translation sample we can see that two sentences have slight grammatical errors and give the impression of being written by a second language English speaker. Some words have been replaced by synonyms but over all, the translated review still conveys the original message, see Table VIII and Table IX.

TABLE VIII
EXAMPLE OF AN ORIGINAL AMT REVIEW

We chose to stay at the Hilton Chicago because it was in such a centralized location- everything that our family wanted to do in town was located so close! What I didn't expect was for the beds to be so comfortable. I can't remember when I got a better night's sleep. The staff was very friendly and the hotel grounds were impeccably kept. We'll be returning to the Hilton Chicago the next time we're in town!

TABLE IX
THE SAME REVIEW TRANSLATED FIRST TO SWEDISH, THEN RUSSIAN AND THEN BACK TO ENGLISH

We decided to stay at the Hilton Chicago, because it was so central to everything that our family would like to do in the city was located so close! What I thought was for the bed to be so convenient. I can not remember when I sleep better at night. The staff was very friendly and the hotel grounds were immaculately kept. We will return to Hilton Chicago next time we are in town!

A diagram of the accuracy as a function of the number of features, for the AMT and CSI data sets, can be seen in Figure 2.

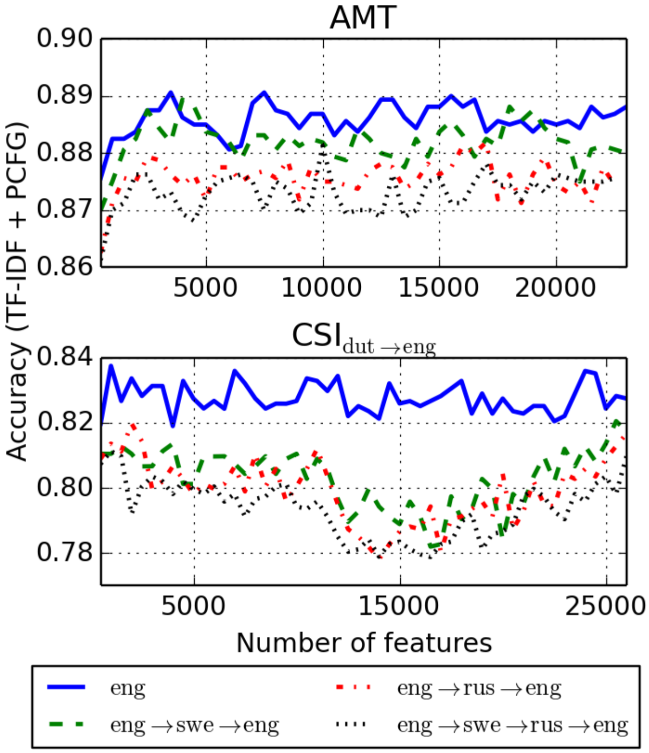

Fig. 2. Accuracy for the different translations on AMT and CSI$_{dut \rightarrow eng}$ using TF-IDF together with PCFG features.

The Tables X and XI summarize the best performance measurements for each of the data sets and translations.

TABLE X
BEST CLASSIFIER APPLIED ON AMT TRANSLATIONS

Translation	A	F
eng	89.06	89.61
eng→swe→eng	88.94	88.46
eng→rus→eng	88.19	86.51
eng→swe→rus→eng	88.12	86.83

TABLE XI
BEST CLASSIFIER APPLIED ON CSI$_{DUT \rightarrow ENG}$ TRANSLATIONS

Translation	A	F
eng	83.74	84.14
eng→swe→eng	82.05	82.20
eng→rus→eng	81.97	77.45
eng→swe→rus→eng	81.13	77.55

VI. DISCUSSION

A. Feature Exploration

Figure 1 depicts the classification model performance in relation to the chosen features when applied to the AMT$_+$ data set. The same pattern occurs for the other data sets as well. The lowest performance is obtained by using PCFG without the lexical rules. Hence, we conclude that there is less information regarding deceptiveness in the constructed rules in the parse tree as in the used words. Even though non-lexicalized PCFG performed worst, the accuracy still reached around 75% for AMT$_+$. It is also clear that for non lexical rules, the accuracy is higher when the grandparent node is included. As a base line comparison [7] did a manual tagging of the AMT$_+$ reviews and the best human judge reached an accuracy of 61,9%, which is lower than what we get with PCFG.

The second worst features are the ones generated using word2vec. Generally, the classification performs worse with a smaller number of features, especially when the data is unstemmed and includes the stop words. The best preprocessing seems to be to remove stop words which also is in line with previous work by [12]. The hypothesis is that stop words increase the noise and make the coordinates of non stop words less informative.

The best feature types are PCFG where lexical nodes are included and TF-IDF. They are performing quite similar regardless of the data set. The reason for this might be that with TF-IDF we are building features where the frequency of every word is used. The same goes for PCFG which also is encoded with TF-IDF. But for PCFG we also have information of which rules occur in the parse tree of the sentence.

In an effort to boost the performance we combined TF-IDF with setting P_1 and PCFG with setting R_2.

In Table XII a summary of the best classifier results for all data sets and comparison with the best accuracy results others have obtained with the same data sets can be seen.

For the AMT$_+$ data set, the work which obtained 91.2%, i.e., [8], used the same feature types as we did, but there is no mentioning how the data was preprocessed or the features optimized which might indicate that our preprocessing or feature selection approach gave us a slight edge.

TABLE XII
SUMMARY OF OUR AND OTHERS' BEST ACCURACY (A)

Data set	Our best A	Others best A
AMT$_+$	**91.5**	91.2 [8]
AMT$_-$	**90.1**	86.0 [9]
AMT$_{\text{TO+}}$	**89.1**	86.0 [9]
AMT$_{\text{TO-}}$	**91.0**	88.4 [9]
CSI$_{\text{dut}\rightarrow\text{eng}}$	**84.7**	72.2 [10]

In [9], the best accuracy on AMT$_-$ was 86%. We managed to improve the accuracy with our model to 90.1%. Our increased accuracy indicates that our feature types work better than standard n-gram for this type of classification and the same conclusion is strengthened by the accuracy of AMT$_{\text{TO+}}$ and AMT$_{\text{TO-}}$.

The biggest accuracy increase has been made on the CSI data set where the accuracy 72.2% reported by [10] has been improved by us to 84.7%. This might be a result of the increased data set size but also because of the more informative feature types.

We also tested how well the model performed when we mixed AMT and CSI$_{\text{dut}\rightarrow\text{eng}}$ and the result is shown in Table VI. The model performs well classifying data from both data sets while trained on both. The best accuracy obtained is slightly higher than for CSI$_{\text{dut}\rightarrow\text{eng}}$ alone but lower than the accuracy obtained for the different AMT data sets. This indicates that it is harder to build a model adapted to several data sets, probably because the data sets, despite both being about reviews, still are different from each other.

B. Classification Model Transferability

From the performance measurements in Table VII, we can make the conclusions that our trained classifier is not suitable for detecting whether reviews from another data set are deceptive or not. The highest accuracy for each train and test combination is not higher than 54%, which is only slightly better than chance indicating that the models become overfitted. Another reason for the low accuracy can be the linguistic awareness of the students which wrote the the CSI reviews. If they knew of the purpose for the reviews they might have been extra aware of their linguistic style and unconsciously put energy in hiding it. It is worth mentioning that even though the accuracies are low in both cases, the case where we train on CSI$_{\text{dut}\rightarrow\text{eng}}$ and test on AMT, we achieve an F-score twice as big as the other case. This was because of the four times higher recall. To achieve a more versatile model a larger training data set, preferably composed from different sources, is probably needed.

C. Translation Impact

Using the "untranslated" results as a base line Figure 2 illustrates that the classification model performs worse on translated test data and that the number of translations have a direct impact on the accuracy. The Swedish translation decreases the accuracy less than the Russian translation and the three-way translation drops the accuracy the most. We conclude that this is because of the linguistic similarities between English and Swedish and the differences with Russian. Even though this trend is not as clear with the CSI data set as with the AMT data set, the same pattern is seen. Machine translation was used by [17] to try and mask authors' stylometric patterns. They concluded that even though they got an accuracy drop between 15% and 35% machine translation was not enough to completely anonymize authors. Our accuracy drop was not nearly as large as the one they got and our purpose with the translation was not the same as theirs but one could still expect a bigger accuracy drop for our application. The reason that we did not see such a drop could be that the machine translators simply have gotten better with time. Another conclusion one can make from this is that multiple way translation is not a viable attack method as the precision robustness is too stable.

VII. CONCLUSIONS

In this work we have experimented with different feature types, preprocessing, settings and data set manipulation to explore detection performance variations of deceptive reviews. As stated in the introduction we have focused on three related research questions to achieve the overarching goal of detecting deceptive reviews.

To study the impact of different features (the first research question) we utilized a linear SVM model and tested it with TF-IDF, word2vec and PCFG in various combinations on different data sets. By using SVM together with TF-IDF (unstemmed with stop words) and PCFG (with lexical rules) results on par with or an improvement on all of the to us known result on the same data sets was obtained.

For the second research question, testing a trained model on data sets for other product types than what they have been trained for, we observed that the accuracy of the trained classifiers is drastically reduced. A result that, although not very surprising, allows us to conclude that with this type of classifier, features and amount of training data there is no general learning of deceptive reviews. Something that is probably desirable in a real world application.

The third research question's purpose is to see whether the classifier might be tricked by an obfuscation of the reviews through machine translation back and forth trough linguistically different languages. The reasoning behind being that machine translation is a method that has been used for blurring authors' stylometric patterns (which is useful for anonymization purposes) [15], [17]. The results of the machine translation experiments showed that there is a slight accuracy drop in the range of a couple of percent. Thus, we conclude that translations are not a viable attack method to avoid detection of deceptive reviews.

ACKNOWLEDGEMENTS

This work was supported by the European Union's Horizon 2020 research and innovation action program under grant agreement no. 832921.

References

[1] C. Avery, P. Resnick, and R. Zeckhauser, "The market for evaluations," *American Economic Review*, pp. 564–584, 1999.

[2] Gartner, "Gartner says by 2014, 10-15 percent of social media reviews to be fake, paid for by companies," http://www.gartner.com/newsroom/id/2161315, 2012, accessed: 2016-06-01.

[3] R. Murphy, "Local consumer review survey — online reviews statistics trends," Bright Local, 2018.

[4] TripAdvisor, "2019 tripadvisor review transparency report," 2019.

[5] C. Dellarocas, "Strategic manipulation of internet opinion forums: Implications for consumers and firms," *Management science*, vol. 52, no. 10, pp. 1577–1593, 2006.

[6] K.-H. Yoo and U. Gretzel, "Comparison of deceptive and truthful travel reviews," *Information and communication technologies in tourism 2009*, pp. 37–47, 2009.

[7] M. Ott, Y. Choi, C. Cardie, and J. Hancock, "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*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 309–319. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002512

[8] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012, pp. 171–175.

[9] M. Ott, C. Cardie, and J. Hancock, "Negative deceptive opinion spam." in *HLT-NAACL*, 2013, pp. 497–501.

[10] B. Verhoeven and W. Daelemans, "CLiPS stylometry investigation (csi) corpus: A dutch corpus for the detection of age, gender, personality, sentiment and deception in text." in *LREC*, 2014, pp. 3081–3085.

[11] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 632–640.

[12] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *Cognitive Informatics Cognitive Computing (ICCI*CC), 2015 IEEE 14th International Conference on*, July 2015, pp. 136–140.

[13] F. Johansson, L. Kaati, and A. Shrestha, "Time profiles for identifying users in online environments," in *Proc. 1st Joint Intelligence and Security Informatics Conference :*, 2014, pp. 83–90.

[14] F. Johansson, L. Kaati, and A. Shresta, "Timeprints for identifying social media users with multiple aliases," *Security Informatics*, vol. 4, no. 1, pp. 1–11, 2015.

[15] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Security and Privacy (SP), 2012 IEEE Symposium on*, May 2012, pp. 461–475.

[16] M. Tsikerdekis and S. Zeadally, "Multiple account identity deception detection in social media using nonverbal behavior," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 8, pp. 1311–1321, Aug 2014.

[17] S. Afroz, M. Brennan, and R. Greenstadt, "Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 3, pp. 12:1–12:22, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2382448.2382450

[18] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70. [Online]. Available: http://dx.doi.org/10.3115/1118108.1118117

[19] B. Yu, "An evaluation of text classification methods for literary study," Ph.D. dissertation, Champaign, IL, USA, 2006, aAI3250350.

[20] A. Rajaraman and J. Ullman, *Mining of massive datasets*. Cambridge University Press Cambridge, 2012, vol. 1.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[22] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[23] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[24] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ser. ACL-44. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 433–440. [Online]. Available: http://dx.doi.org/10.3115/1220175.1220230

[25] A. Moh'd A Mesleh, "Chi square feature extraction based svms arabic language text categorization system," *Journal of Computer Science*, vol. 3, no. 6, pp. 430–435, 2007.

[26] C. Clapham and J. Nicholson, *The concise Oxford dictionary of mathematics*. OUP Oxford, 2009.

[27] J. Fernquist, "Detection of deceptive reviews: using classification and natural language processing features," 2016.