Detecting deceptive reviews using Argumentation

Oana Cocarascu
Imperial College London
United Kingdom
SW7 2AZ, London
oc511@imperial.ac.uk

Francesca Toni
Imperial College London
United Kingdom
SW7 2AZ, London
f.toni@imperial.ac.uk

ABSTRACT

The unstoppable rise of social networks and the web is facing a serious challenge: identifying the truthfulness of online opinions and reviews. In this paper we use Argumentation Frameworks (AFs) extracted from reviews and explore whether the use of these AFs can improve the performance of machine learning techniques in detecting deceptive behaviour, resulting from users lying in order to mislead readers. The AFs represent how arguments from reviews relate to arguments from other reviews as well as to arguments about the goodness of the items being reviewed.

1. INTRODUCTION

Nowadays the decision of purchasing a specific product or service is often based on online reviews. However, the authenticity and truthfulness of these reviews is not guaranteed and content communities, review and news websites are susceptible to deceptive content. Different deception strategies exist: falsification (contradictions/ lies), exaggeration (superlative information), omission (hiding information) and misleading information (irrelevant information/ topic changes) [1]. Opinions expressed in online forums or e-commerce websites attract and influence potential customers. It has been found that 87% people change their purchase decision after reading positive reviews whereas 80% people changed their decision of purchasing a product after reading a negative review [18]. However, the human accuracy in detecting deceptive opinions is only 61.9% [32]. Although they are not considered to be malware, deceptive reviews can pose risks for security and privacy by persuading potential customers to buy a company's product/ service given positive reviews or discouraging customers from purchasing when faced with negative reviews. Deception technology has been acknowledged as an emerging security technology[1]. Whilst threat deception can be used in some settings, in this paper we focus on a very specific type of deception, namely deceptive reviews.

Most work on detecting deceptive reviews uses machine learning techniques and features extracted by Natural Language Processing (NLP) (e.g. see [13]). We propose new features, obtained through (special forms of) Argumentation Frameworks as understood in AI (see [3, 39] for overviews), and experiment with their use for detecting deceptive reviews by several machine learning techniques in two domains. In particular we use *Abstract Argumentation Frameworks* (AAFs) [15] and *Bipolar Argumentation Frameworks* (BAFs) [11]. These AFs represent dialectical (attack for AAFs and attack/support for BAFs) relationships between arguments, with arguments seen simply as abstract entities, and are equipped with semantics/algorithms to compute acceptability [15, 11] or dialectical strength [38] of arguments, given the relationships amongst them.

Argument Mining is a relatively new research area which involves, for instance, the automatic detection of arguments in text, of argument components, as well as of relations between arguments (e.g. see [33, 37, 27] for overviews). In our approach, we mine arguments and relationships between them from reviews to get AFs. Then, the strengths of arguments in the AFs we mine contribute new *argumentative features* for standard machine learning classifiers. We use two methods for Argument Mining. The first method uses sentiment analysis [34] to construct an AAF from a set of reviews whereas the second method uses relation-based Argumentation Mining [10] alongside sentiment analysis to mine a BAF from a set of reviews. The second method associates arguments to (noun-level) topics in reviews, whereas in the first method arguments are topic-independent.

Our new argumentative features are calculated using the strength of arguments in AFs to capture the impact of each review on determining how good/bad the item (product or service) is with respect to all reviews about that item. Thus, these argumentation features can be seen as adding a semantic layer to the analysis of deceptive behaviour in reviews on top of the syntactic analysis given by standard NLP when using machine learning techniques. Our approach can also be seen as integrating argumentation and machine learning, in the spirit, for instance, of [28, 19, 9, 8], but in a different context (deception detection) and using a novel methodology (argumentative features).

In order to test the usefulness of our novel argumentative features to determine deceptive reviews, we experiment with various machine learning classifiers, using the gold standard consisting of hotel reviews of 20 Chicago hotels [31] and

[1]http://thevarguy.com/network-security-and-data-protection-software-solutions/090115/gartner-deception-key-emerging-security-tech

restaurant reviews [25]. We show experimentally that, for a number of classifiers, using argumentative features yields no change or better results in classifier performance. In the case of the AAF-based argumentative features, we obtain an improvement of 1.5% accuracy for the hotel dataset and 2.25% for the restaurant dataset when compared to the baseline. In the case of the BAF-based argumentative features, we obtain an improvement of 3.5% accuracy for the hotel domain and 4% for the restaurant domain when compared to the baseline. In the experiments, to determine both AAF- and BAF-based argumentative features, we use an off-the-shelf sentiment analysis classifier. To determine BAF-based argumentative features, we train a relation-based Argument Mining classifier, achieving 96.19% F_1 for determining support/attack/neither relationships between sentences.

The remainder of the paper is organised as follows. We discuss related work and give an overview of AFs and datasets used in this study in Section 2. In Section 3 we describe our approach to extracting arguments from reviews and building AFs. In Section 4 we define the argumentative features drawn from these AFs. In giving the argumentative features, we also present a method for calculating argument strength. We show the results of our experiments in Section 5 where we also report some novel qualitative findings about differences between deceptive reviews and truthful reviews. We conclude the paper and propose directions for future work in Section 6.

2. BACKGROUND

2.1 Related work

Review spam detection or deception detection has recently received a great deal of attention. Much of the previous work on detecting deceptive reviews focused on detecting either reviews (e.g. opinion spam) [32, 42, 18] or the actors of deception (e.g. deceptive spammers) [26, 30]. Some existing work looks at identifying reviews written by the same person but under different names [40]. Given that the majority of users write a single review, others focus on identifying singleton deceptive reviews e.g. using multi-scale multidimensional time series anomalies based on the assumption that a large number of deceptive reviews are given in a short period of time and are correlated to the rating [43]. Some other existing work focuses on detecting single review spammers [26] and group review spammers [30]. Our work focuses on determining whether single reviews are deceptive.

Different forms of machine learning have been used in the literature to detect deceptive behaviour, notably unsupervised [29], semi-supervised [18] and supervised [32, 25, 31, 42] techniques. Different techniques use different features. These can be divided into two main groups: features related to the review and features related to the reviewer [24, 21]. Some previous work singles out quantity, specificity, diversity, non-immediacy, as well as task specific features such as affect, expressivity, complexity, uncertainty and informality [17, 44]. A more detailed overview of the machine learning techniques and features used to detect review spam is given in [13]. Our experiments use Logistic Regression, Naïve Bayes and Random Forests classifiers, and a combination of features from the literature to serve as baseline.

Existing Argument Mining approaches focus on different tasks, including identifying argumentative sentences, argument components and the structure of arguments (e.g. claims and premises), and relations between arguments (e.g. support/attack). Previous studies proposed various classifications of (parts of argumentative) text, such as: callout and target [20], unverifiable, verifiable (non-experiential and experiential) and non-argumentative [35], claims and premises [41], A (explicitly attacking an argument), a (vaguely/ implicitly attacking an argument), N (no use of argument), s (vaguely/ implicitly supporting an argument) and S (explicitly supporting an argument) [5]. Supervised algorithms such as Support Vector Machines (e.g [33, 35, 5]), Naïve Bayes (e.g. [33]) and Maximum Entropy (e.g. [33]) have been used for this task. [6, 7] identify arguments within text and determine relations between these arguments using textual entailment. In [23], the presence of discourse indicators gives the relations between propositions. The argumentative structure is then constructed and topic similarity is used to connect propositions that were overlooked based on the argumentation scheme used. Some studies focused on identifying argumentative relations (attack, support, neither) by classifying pairs of sentences (e.g. [10]). For overviews of approaches in Argument Mining see, for instance, [33, 37, 27]. In our experiments we perform topic-dependent Argument Mining, using Random Forests, trained on a mixture of corpora from [12, 22, 10].

2.2 Argumentation Frameworks

(Abstract) Argumentation Frameworks (AAFs), introduced by Dung [15], are pairs consisting of a set of arguments and a binary relation between arguments, representing attacks. Formally, an AAF is any $\langle AR, attacks \rangle$ where $attacks \subseteq AR \times AR$.

Bipolar Argumentation Frameworks (BAFs) extend AAFs by considering two independent binary relations between arguments: attack and support [11]. Formally, a BAF is any $\langle AR, attacks, supports \rangle$ where $attacks \subseteq AR \times AR$ and $supports \subseteq AR \times AR$.

In this paper we use both AAFs and BAFs to define our argumentative features.

Semantics of AAFs and BAFs are standardly defined in terms of "winning" sets of arguments, where, for example, given $\langle AR, attacks \rangle$, $S \subseteq AR$ can be deemed to be "winning" if it is *admissible* namely (by lifting the attack relation to sets in the standard way) S does not attack itself and attacks all arguments attacking it.

Alternatively, as in this paper, semantics of AAFs and BAFs can be defined in terms of a notion of *strength* (e.g. [2]) namely a function from AR to (a suitable subset of) the real numbers. Like in [2] this strength can be obtained from a given *base score* of arguments defined as a function $BS : AR \rightarrow [0, 1]$, a function $\mathcal{F}$ for aggregating the strengths of arguments and a function $\mathcal{C}$ for combining the base score of arguments with the aggregated score of their attackers and supporters. We define appropriate BS, $\mathcal{F}$, $\mathcal{C}$ and *strength*, for our purposes, in Section 4.1.

2.3 Datasets

The gold standard for deceptive reviews consists of positive and negative hotel reviews of 20 Chicago hotels [31] extended more recently to include deceptive reviews written by domain experts (employees) and Amazon Mechanical Turkers, and truthful reviews written by customers from three domains: hotels, restaurants and doctors [25]. Studies have

focused on detecting deceptive hotel reviews [32], identifying positive and negative deceptive hotel reviews [31] and cross-domain deception on the more recent data set [25].

The hotel dataset used in this paper consists of 1600 positive and negative reviews from this gold standard about 20 Chicago hotels: 400 truthful positive reviews from TripAdvisor, 400 truthful negative reviews from 6 online review websites, 400 deceptive positive reviews and 400 deceptive negative reviews from Turkers [31]. The restaurant dataset used in this paper is the one given in [25].

3. MINING AFS FOR DETECTING DECEPTION

We use two methods to extract AFs from reviews. The first method (Section 3.1) generates an AAF from a set of reviews; the arguments in this AAF are topic-independent. The second method (Section 3.2) generates a BAF from a set of reviews; the arguments in this BAF are topic-dependent. Both methods use an off-shelf sentiment analysis classifier to determine (attack/support, as applicable) relations between arguments. The second method also uses relation-based Argumentation Mining. The concrete implementation choices for experimentation, for both sentiment analysis and relation-based Argumentation Mining, will be discussed in Section 5.

3.1 Building a topic-independent AAF

The arguments in these AAFs include two special arguments, referred as G (for 'good') and B (for 'bad'). Additionally, each AAF includes a number of arguments extracted from the reviews under consideration. We assume that each such argument is contained in a sentence from a review. Thus, each review is mapped to one or more such arguments. To determine the attack relation in our AAF, we use the polarity of arguments extracted from the reviews as follows: argument a from a review attacks G (B) if the sentiment of a is - (+ respectively).

For example, consider the following reviews about some hotel H:

r_1: *'It had nice rooms but terrible food.'*

r_2: *'Their service was amazing and we absolutely loved the room. They do not offer free Wi-Fi so they expect you to pay to get Wi-Fi...'*

From r_1 we extract the following arguments, with polarity as indicated:

a_{11}: *It had nice rooms* (+)
a_{12}: *(It had) terrible food* (-)

whereas from r_2 we obtain:

a_{21}: *service was amazing* (+)
a_{22}: *absolutely loved the room* (+)
a_{23}: *they do not offer free Wi-Fi so they expect you to pay to get Wi-Fi* (-) [2]

The AAF $\langle AR, attacks\rangle$ obtained from reviews r_1 and r_2 thus has

$AR = \{G, B, a_{11}, a_{12}, a_{21}, a_{22}, a_{23}\}$ and
$attacks = \{(a_{12}, G), (a_{23}, G), (a_{11}, B), (a_{21}, B), (a_{22}, B)\}$

[2]Note that we use components of argumentative sentences to stand for the full sentences. For example, a_{11} stands for *'The hotel was good as it had nice rooms'*. This is in the spirit of AA, where arguments can be anything.

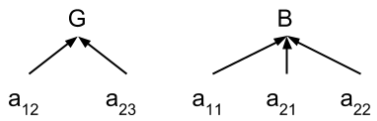

Figure 1: AAF obtained from r_1,r_2 in Section 3.1.

shown graphically, as conventional in the Computational Argumentation literature, in Figure 1[3].

3.2 Building a topic-dependent BAF

In building a topic-dependent BAF from a set of reviews, we first identify topics mentioned in the reviews, at noun-level (e.g. topics are *room, food, service, Wi-Fi* given the example reviews in Section 3.1). We then identify the sentences/arguments, as in Section 3.1, but related to these topics, and finally determine the relations between the arguments. The arguments in the constructed BAFs include a single special argument G (for 'good') as well as a special argument G_t per topic t identified (for 'good as far as t is concerned'), such that each G_t supports G.

In order to determine the relations between arguments related to topic t drawn from reviews and the special argument G_t, we assume that a newer argument (with respect to time) can either support, attack, or neither support nor attack a previous argument or G_t, but not vice versa. If an argument a_t, related to topic t, does not support or attack another argument related to t from the same or some other review, as determined by relation-based Argument Mining, then this argument a_t will either support or attack G_t, according to its polarity as determined by sentiment analysis. This 'timeline' approach of constructing the BAF is a limitation of our model which will be addressed in future work.

For example, given reviews r_1 and r_2 from Section 3.1. we obtain the BAF $\langle AR, attacks, supports\rangle$ with

$$\begin{aligned} AR = &\{G, G_{room}, G_{food}, G_{service}, G_{Wi-Fi}, \\ &a_{11}, a_{12}, a_{21}, a_{22}, a_{23}\}, \\ attacks = &\{(a_{12}, G_{food}), (a_{23}, G_{Wi-Fi})\} \\ supports = &\{(a_{22}, a_{11}), (a_{11}, G_{room}), \\ &(a_{21}, G_{service}), (G_{room}, G), (G_{food}, G), \\ &(G_{service}, G), (G_{Wi-Fi}, G)\} \end{aligned}$$

shown graphically in Figure 2 (where edges labelled - represent attacks and edges labelled + represent supports).

4. FROM AFS TO ARGUMENTATIVE FEATURES

In order to detect deceptive reviews, in addition to standard features used in previous studies, we associate argumentative features to each review, representing the impact of the review on how good/bad (when using topic-independent AAFs) or how good (when using topic-dependent BAFs) an item (e.g. hotel or restaurant) is with respect to all reviews

[3]When representing an AAF as a graph, nodes represent arguments and edges represent attacks.

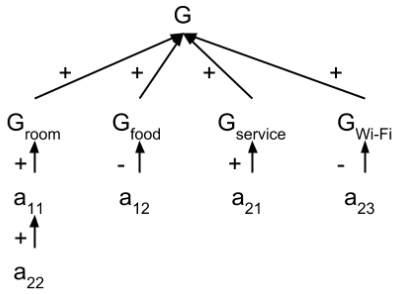

Figure 2: Topic-dependent BAF obtained from r_1,r_2 in Section 3.2.

about that item. These new features are obtained from measuring the strength of arguments in the AF built from all reviews related to the chosen item and in the AF built from all reviews for that item except the one whose impact we aim at determining. We define the notion of strength we use in our experiments in Section 4.1, and the argumentative features in Section 4.2.

4.1 Calculating argument strength

The AFs obtained from sets of reviews, as described in Section 3, are, by construction, guaranteed to be in the restricted form of sets of trees. Note that these trees may have any (finite) breadth, but have necessarily depth 1 in the case of AAFs, whereas they can be of any depth in the case of BAFs, as determined by the relations between arguments extracted from reviews.

Given that the AFs are (sets of) trees, the strengths of arguments in these AFs can be calculated recursively as follows in terms of a strength aggregation function $\mathcal{F}$ and combination function $\mathcal{C}$ (modifying the corresponding notions in [38], also defined for trees).

Our strength aggregation function $\mathcal{F}$, given n arguments with strengths $v_1, ..., v_n$, is defined as:

$$\mathcal{F}(v_1, ..., v_n) = \begin{cases} 0 & n = 0 \\ 1 - \log \prod_{i=1}^{n} (|1 - v_i|) & n > 0 \end{cases}$$

Here and below, we apply the logarithm to avoid underflow in the implementation.

Our combination function $\mathcal{C}$, for an argument with base score v_0, attackers with strengths $v_1, ..., v_n$ (for $n \geq 0$, $n = 0$ amounts to the argument having no attacker) and supporters with strengths $v'_1, ..., v'_m$ (for $m \geq 0$, $m = 0$ amounts to the argument having no supporters) is defined as follows, for $v_a = \mathcal{F}(v_1, ..., v_n)$ and $v_s = \mathcal{F}(v'_1, ..., v'_m)$: $\mathcal{C}(v_0, v_a, v_s) =$

$$\begin{cases} v_0 & if\ v_a = v_s \\ v_0 - log(v_0 * |v_s - v_a|) & if\ v_a > v_s \\ v_0 + log((1 - v_0) * |v_s - v_a|) & if\ v_a < v_s \end{cases}$$

Finally, for any argument $a \in AR$ with $BS(a) = v_0$ and n attackers with strengths $v_1, \ldots, v_n$ and m supporters with strengths $v'_1, \ldots, v'_m$ we define
$strength(a) = abs(\mathcal{C}(v_0, \mathcal{F}(v_1, ..., v_n), \mathcal{F}(v'_1, ..., v'_m)))$.
We take the absolute value in order to guarantee that the two new features per review we obtain (see below) are positive, as some classifiers require.

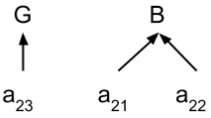

Figure 3: AAF obtained from removing (arguments from) r_1 (see Section 4.2).

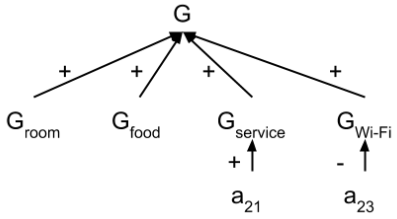

Figure 4: BAF obtained from removing (arguments from) r_1 (see Section 4.2).

Note that if the AF is an AAF, then v_s, for any argument, is 0, as there are no supporters for any argument. For illustration, consider the AAF extracted earlier from reviews r_1, r_2 (see Figure 1). Assume a base score function BS such that, for all $a \in AR$, $BS(a) = 0.5$ (we will use this same base score for all arguments in our experiments). Then $strength(G)$ is

$$abs(\mathcal{C}(v_0, \mathcal{F}(a_{12}, a_{23}), 0)) =$$
$$abs(\mathcal{C}(0.5, 1 - log(1 - 0.5) - log(1 - 0.5), 0)) =$$
$$abs(\mathcal{C}(0.5, 2.38629436112, 0)) = 0.323405.$$

Similarly, $strength(B) = 0.068399$.

4.2 Argumentative features

The strength of G/B can be seen as a measure of how good/bad the item is deemed to be according to the reviews under consideration. The impact of review r is then given by the absolute difference between the measure of how good/bad the hotel is deemed to be given all reviews R and how good/bad, respectively, it is deemed to be given R$\backslash\{r\}$.

In our example, if R=$\{r_1, r_2\}$, to calculate the impact of r_1 requires removing from our earlier AAF all arguments from r_1, giving $AR' = \{G, B, a_{21}, a_{22}, a_{23}\}$,
$attacks' = \{(a_{23}, G), (a_{21}, B), (a_{22}, B)\}$ as shown in Figure 3. Then $strength(G) = 0.666558$ and $strength(B) = 0.323405$. Thus, the new features indicating the impact of r_1 are $F_G = 0.343153$, $F_B = 0.255006$.

Given instead the BAF in Figure 2, the BAF obtained by removing all arguments from r_1 is shown in Figure 4. The strength of G can be seen as a measure of how good the product is deemed to be according to the reviews under consideration. Note that in this case we also remove argument a_{22} from review r_2 because a_{22} cannot be connected to any previous argument from any review under consideration (see Section 3.2 on how the BAF was built).

5. RESULTS AND DISCUSSION

Category	Features
Personalization	nr self references nr 2nd person pronouns nr other references nr group pronouns
Quantity	nr sentences nr words nr nouns nr verbs
Complexity	avg sentence length avg word length
Diversity	lexical
Uncertainty	nr modal verbs nr modifiers

Table 1: Features and the associated category. (*nr* stands for number and *avg* for average)

We evaluate the performance of several algorithms with and without argumentative features, using the gold standard for deception detection (see Sections 5.2–5.4). The algorithms tested are Logistic Regression (LR), Naïve Bayes (NB) and Random Forests (RF). We use 5-fold cross-validation as Ott et al. [32]. We report accuracy (A) and F_1. All algorithms use standard features, obtained by standard Natural Language Processing (NLP). Standard NLP techniques are also used to determine topics and arguments from reviews, and thus identify part of the AFs that contribute to the argumentative features. The NLP components of our experiments are described in Section 5.1.

As part of our evaluation, since the argumentative features are based on sentiment analysis, we also report, in Section 5.5, results when using, instead of argumentative features measuring the impact of reviews on the strength of (arguments for) an item being good/bad, sentiment analysis features measuring the sentiment score of the item and sentiment analysis features measuring the impact of reviews on the sentiment score of the item. Finally, in Section 5.6, we report on some qualitative findings from the experiments.

5.1 Use of NLP techniques

Our first step is splitting each review into sentences with a pre-trained tokenizer for English from *nltk* [4]. Sentences containing *'but'* or *'although'* are split since generally the phrases before and after these separators express different sentiments (e.g. *'The staff was nice but the room was messy'* results in two sentences with different sentiments). In this first step we also extract features used previously in studies of deception (see Section 2.1). These features are the result of Part-Of-Speech (POS) tag analysis using *nltk* and are summarised in Table 5.1. Additionally, we include *tf-idf* (term frequency-inverse document frequency) features obtained from all reviews using *scikit-learn* [36]. To calculate these, we use the lemmas obtained by analysing the lowercase form of words and their POS tag.

The sentiment polarity of each argument is determined using sentiment analysis from the *pattern.en* module [14] which uses a lexicon of frequently used adjectives in product reviews annotated with scores for sentiment polarity.

5.2 Baseline - NLP features

Table 2 shows the results of the three classifiers on the two datasets we use (see Section 2.3) using standard NLP features as summarised in Section 5.1 and represents our baseline.

Dataset	Hotel		Restaurant	
Classifier	**A%**	$\boldsymbol{F_1}$%	**A%**	$\boldsymbol{F_1}$%
LR	85.19	85.3	80.5	80.94
NB	73.88	74.36	74	77.22
RF	76.25	73.92	69	67.56

Table 2: Classifiers' performance without argumentative features (baseline).

5.3 Topic-independent AAF argumentative features

Table 3 shows the results of the three classifiers using the argumentative features in Section 3.1. Including these argumentative features results in a slight improvement (0.06%-1.5%) in accuracy for LR and RF and no change in performance in the case of NB on the hotel dataset. On the restaurant dataset, including these argumentative features results in an improvement (0.5%-2.25%) in accuracy for LR and RF and no change in performance in the case of NB when compared to the baseline.

Dataset	Hotel		Restaurant	
Classifier	**A%**	$\boldsymbol{F_1}$%	**A%**	$\boldsymbol{F_1}$%
LR	85.25	85.35	81	81.48
NB	73.88	74.36	74	77.22
RF	77.75	75.94	71.25	68.68

Table 3: Classifiers' performance including (AAF) argumentative features.

5.4 Topic-dependent BAF argumentative features

Determining relations between any pair of sentences/ arguments can be viewed as a three-class problem, with classification labels $L = \{attack, support, neither\}$. We developed a classification model for determining relations between arguments using the Araucaria corpus [12] from AIFdb [22], a database of argument structures (where node type CA represents attack and node types RA/TA represent support), and a corpus extracted from news with three classes (*support*, *attack*, *neither*) [10]. The classification model is obtained using Random Forests, using the features shown in Table 4. In particular, for the 'combined semantic and syntactic' feature, we use two similarity measures between words: *path* represents the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy and *lch* represents the Leacock-Chodorow similarity, namely the shortest path between the senses divided by double the maximum depth in the taxonomy in which the senses occur. We used a dataset of more than 20000 pairs of sentences, covering topics such as UKIP, opinions about movies, technology and politics, where *attack* relations represent 27% of the dataset, *support* relations represents 43% of the dataset and *neither* relations represent 30% of the dataset. Using stratified cross validation (so that each fold is a good representative of the whole), Random Forests yielded F_1 96.19%.

Using a single argumentative feature drawn from topic-dependent BAFs yields an improvement in accuracy for RF of 3.5% on the hotel dataset and 4% on the restaurant

Feature	Detail
number of words	for each sentence
avg word length	for each sentence
sentiment polarity	for each sentence
Jaccard similarity	size of the intersection of words in sentences compared to the size of union of words in sentences
Levenshtein distance	count of replace and delete operations required to transform one sentence into the other
word order	normalized difference of word order between the sentences
Malik	sum of maximum word similarity scores of words in same POS class normalized by sum of sentence's lengths (path and lch)
combined semantic and syntactic	linear combination of semantic vector similarity and word order similarity (path and lch)

Table 4: Overview of features used in determining relations between pairs of sentences.

dataset, with slight improvements for LR and NB compared to the baseline. All performances we obtain are comparable with previous studies where results on the gold standard vary between 65% and 89.8%. [13].

Dataset	Hotel		Restaurant	
Classifier	A%	F_1%	A%	F_1%
LR	85.44	85.55	81	81.43
NB	73.94	74.41	74.25	77.48
RF	79.81	78.21	73	70.76

Table 5: Classifiers' performance including (BAF) argumentative features.

5.5 Additional experiments

Table 6 shows results when including features related to sentiment polarity (a feature for positive polarity and a feature for negative polarity). Concretely, for each review we identify the sentences/arguments with positive sentiment polarity and the sentences/arguments with negative sentiment polarity drawn from that review. The positive/negative polarity score of a review is the average sum of the polarity scores of positive/negative sentences/arguments for the review. Thus, we obtain two new features, representing the average positive/negative polarity of the review, respectively. Including features related to sentiment polarity results in an increase in accuracy of 2% for RF on the hotel dataset and a decrease in performance for each classifier tested on the restaurant dataset compared to the baseline.

Dataset	Hotel		Restaurant	
Classifier	A%	F_1%	A%	F_1%
LR	85.31	85.37	79.25	80.01
NB	73.87	74.4	73.5	76.87
RF	78.44	76.59	68.5	64.7

Table 6: Classifiers' performance using sentiment polarity features rather than argumentative features.

Table 7 shows the results when including two new features related to the impact each review has on sentiment polarity with respect to all other reviews. More precisely, the two new features represent, for review r, the absolute difference between the average positive (negative) sentiment polarity score given all reviews R and the average positive (negative) polarity score given R$\backslash\{r\}$. Using these two new features, the accuracy is improved by 1.75% for RF for hotels and by 0.5% for restaurants when compared to the baseline.

Dataset	Hotel		Restaurant	
Classifier	A%	F_1%	A%	F_1%
LR	85.62	85.73	80.25	80.89
NB	73.88	74.33	73.75	77.05
RF	78	76.23	70.5	67.28

Table 7: Classifiers' performance using sentiment polarity impact features rather arguments.

Overall, the use of BAF argumentative features gives consistently better performances when used with Random Forests compared to the baseline.

5.6 Qualitative findings

We conducted additional experiments to single out features of deceptive reviews and found that on average deceptive reviews have more modal verbs whilst the use of third person did not appear to be a good separator between deceptive and truthful reviews. These findings are in line with previous findings [44]. We also found that deceptive reviews have more self references (including group self references), as also reported in [25, 32]. This can be attributed to deceitful users trying to make their reviews more realistic. In addition, we found that truthful reviews seem to be longer (in terms of number of words) than deceptive reviews. This suggests that deceptive reviews provide less information. Another difference we noticed is that truthful reviews have more adjectives and adverbs. This can be attributed to the fact that deceptive reviews are written to seem authentic. We also looked at topics that appear in reviews. Truthful reviews tend to mention more topics but these are also discussed in other reviews, suggesting that fake reviews may contain topics not previously reviewed. Since these experiments were run at noun-level, further analysis is required to determine whether these topics are indeed relevant.

6. CONCLUSION AND FUTURE WORK

To detect deceptive reviews, in addition to standard NLP features, we introduce argumentative features that capture semantic information from reviews represented as Argumentation Frameworks (AFs). Our technique combines Argument Mining with evaluation of strength of arguments from AFs extracted from reviews using NLP and sentiment analysis. We show experimentally, for reviews about hotels and restaurants, that including the argumentative features yields no change or better results in classifier performance, with improvement up to 3.5% for the hotel dataset and an improvement up to 4% for the restaurant dataset.

Further experimentation is needed to investigate whether the use of argumentative features extracted by Argument Mining can bring further performance improvements for detecting deceptive reviews. In particular future directions include investigating other Argument Mining techniques for better extraction of arguments and relations, looking at sets

of arguments that are coherent as an overall opinion, a semi-supervised approach to overcome the dependence on datasets, adding context features (bigrams, trigrams) and exploring other notions of strength and computed, rather than given, base scores for arguments. Another approach would be to explore Wordnet hypernyms [16] to cluster features into related topics or incorporate textual entailment, already used to determine semantic interactions between sentences, and specifically whether one text can be inferred from another [6]. We plan to conduct further experiments regarding the topic vs non-topic AFs and more precisely to test the current AFs with a topic AAF and a non-topic BAF respectively. Finally, obtaining the graphical representation of the AF can be used to better understand the label assigned to a review by the classifier. We plan to explore experimentally whether this may be true.

Whilst our work focuses on a very specific type of deception (deceptive reviews), it would be interesting to test whether the method applies to other domains, hence providing a mechanism for detecting deception to guarantee cyber security.

7. REFERENCES

[1] D. S. Appling, E. J. Briscoe, and C. J. Hutto. Discriminative models for predicting deception strategies. In *Proceedings of the 24th International Conference on World Wide Web*, pages 947–952. ACM, 2015.

[2] M. Aurisicchio, P. Baroni, D. Pellegrini, and F. Toni. Comparing and integrating argumentation-based with matrix-based decision support in arg&dec. In *Theory and Applications of Formal Argumentation - Third International Workshop, TAFA*, pages 1–20. Springer, 2015.

[3] P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[4] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[5] F. Boltužić and J. Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58. Association for Computational Linguistics, 2014.

[6] E. Cabrio and S. Villata. Natural language arguments: A combined approach. In *ECAI*, volume 242, pages 205–210. IOS Press, 2012.

[7] E. Cabrio and S. Villata. Detecting Bipolar Semantic Relations among Natural Language Arguments with Textual Entailment: a Study. In *Joint Symposium on Semantic Processing (JSSP)*, 2013.

[8] L. Carstens. *Using Argumentation to improve classification in Natural Language problems*. PhD thesis, Imperial College London, 2016.

[9] L. Carstens and F. Toni. Improving out-of-domain sentiment polarity classification using argumentation. In *IEEE International Conference on Data Mining Workshop*, pages 1294–1301, 2015.

[10] L. Carstens and F. Toni. Towards relation based argumentation mining. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 29–34. Association for Computational Linguistics, 2015.

[11] C. Cayrol and M.-C. Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *Proceedings of the 8th European conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 378–389. Springer Berlin Heidelberg, 2005.

[12] G. R. Chris Reed, Raquel Mochales Palau and M.-F. Moens. Language resources for studying argument. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2008.

[13] M. Crawford, T. Khoshgoftaar, J. Prusa, A. Richter, and H. Al Najada. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(1):1–24, 2015.

[14] T. De Smedt and W. Daelemans. Pattern for python. *Journal of Machine Learning Research*, 13:2031–2035, 2012.

[15] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321 – 357, 1995.

[16] C. Fellbaum. Wordnet and wordnets. In *Encyclopedia of Language and Linguistics*, pages 665–670, Oxford, 2005.

[17] C. M. Fuller, D. P. Biros, D. P. Twitchell, J. K. Burgoon, and M. Adkins. An analysis of text-based deception detection tools. In *12th Americas Conference on Information Systems*, page 418. Association for Information Systems, 2006.

[18] D. H. Fusilier, M. Montes-y-Gómez, P. Rosso, and R. Guzmán-Cabrera. Detecting positive and negative deceptive opinions using pu-learning. *Information Processing & Management*, 51(4):433–443, 2015.

[19] Y. Gao and F. Toni. Argumentation accelerated reinforcement learning for cooperative multi-agent systems. In *ECAI*, pages 333–338. IOS PRESS, 2014.

[20] D. Ghosh, S. Muresan, N. Wacholder, M. Aakhus, and M. Mitsui. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 39–48. Association for Computational Linguistics, 2014.

[21] N. Jindal and B. Liu. Analyzing and detecting review spam. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 547–552. IEEE Computer Society, 2007.

[22] J. Lawrence, F. Bex, C. Reed, and M. Snaith. Aifdb: Infrastructure for the argument web. In *Computational Models of Argument - Proceedings of COMMA*, volume 245, pages 515–516. IOS Press, 2012.

[23] J. Lawrence and C. Reed. Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 127–136. Association for Computational Linguistics, 2015.

[24] F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2488–2493. AAAI Press, 2011.

[25] J. Li, M. Ott, C. Cardie, and E. H. Hovy. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the*

Association for Computational Linguistics, ACL, pages 1566–1576. Association for Computer Linguistics, 2014.

[26] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 939–948. ACM, 2010.

[27] M. Lippi and P. Torroni. Argument Mining: A Machine Learning Perspective. In *The 2015 International Workshop on Theory and Applications of Formal Argument*, pages 163–176. Springer International Publishing, 2015.

[28] M. Možina, M. Guid, J. Krivec, A. Sadikov, and I. Bratko. Fighting knowledge acquisition bottleneck with argument based machine learning. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 234–238. IOS Press, 2008.

[29] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting opinion spammers using behavioral footprints. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, pages 632–640. ACM, 2013.

[30] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web*, pages 191–200. ACM, 2012.

[31] M. Ott, C. Cardie, and J. T. Hancock. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2013.

[32] M. Ott, Y. Choi, C. Cardie, and J. T. 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*, pages 309–319. Association for Computational Linguistics, 2011.

[33] R. M. Palau and M. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[34] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, Jan. 2008.

[35] J. Park and C. Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38. Association for Computational Linguistics, 2014.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[37] A. Peldszus and M. Stede. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, 2013.

[38] A. Rago, F. Toni, M. Aurisicchio, and P. Baroni. Discontinuity-free decision support with quantitative argumentation debates. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*, pages 63–73. AAAI Press, 2016.

[39] I. Rahwan and G. R. Simari. *Argumentation in Artificial Intelligence*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[40] V. Sandulescu and M. Ester. Detecting singleton review spammers using semantic similarity. In *Proceedings of the 24th International Conference on World Wide Web*, pages 971–976. ACM, 2015.

[41] C. Sardianos, I. M. Katakis, G. Petasis, and V. Karkaletsis. Argument extraction from news. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 56–66. Association for Computational Linguistics, 2015.

[42] S. Shojaee, M. A. A. Murad, A. bin Azman, N. M. Sharef, and S. Nadali. Detecting deceptive reviews using lexical and syntactic features. In *13th International Conference on Intellient Systems Design and Applications, ISDA*, pages 53–58. IEEE, 2013.

[43] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 823–831. ACM, 2012.

[44] L. Zhou, J. K. Burgoon, J. F. Nunamaker, and D. Twitchell. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group Decision and Negotiation*, 13(1):81–106, 2004.