Combining Deep Learning and Argumentative Reasoning for the Analysis of Social Media Textual Content Using Small Data Sets

Oana Cocarascu
Imperial College London
Department of Computing
oana.cocarascu11@imperial.ac.uk

Francesca Toni
Imperial College London
Department of Computing
f.toni@imperial.ac.uk

The use of social media has become a regular habit for many and has changed the way people interact with each other. In this article, we focus on analyzing whether news headlines support tweets and whether reviews are deceptive by analyzing the interaction or the influence that these texts have on the others, thus exploiting contextual information. Concretely, we define a deep learning method for relation–based argument mining to extract argumentative relations of attack and support. We then use this method for determining whether news articles support tweets, a useful task in fact-checking settings, where determining agreement toward a statement is a useful step toward determining its truthfulness. Furthermore, we use our method for extracting bipolar argumentation frameworks from reviews to help detect whether they are deceptive. We show experimentally that our method performs well in both settings. In particular, in the case of deception detection, our method contributes a novel argumentative feature *that, when used in combination with other features in standard supervised classifiers, outperforms the latter even on small data sets.*

1. Introduction

The use of social media has become a regular habit for many and has changed the way people interact with each other. In this article, we focus on analyzing whether news headlines support tweets and whether reviews are deceptive by analyzing the interaction or the influence that these texts have on the others, thus exploiting contextual information.

The recent success of deep learning has led to a widespread use of deep neural networks in a number of domains, from natural language understanding to computer vision, that typically require very large data sets (Dean et al. 2012; Krizhevsky, Sutskever, and Hinton 2012; Lecun, Bengio, and Hinton 2015; Silver et al. 2016). In this

Submission received: 13 October 2017; revised version received: 25 May 2018; accepted for publication: 19 August 2018.

doi:10.1162/coli_a_00338

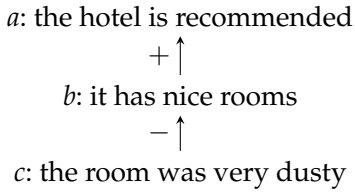


Figure 1
Example BAF. Here, *b* supports *a* (indicated as a + edge) and *c* attacks *b* (indicated as a − edge).

article, we propose a deep learning method to extract relations of *attack* and *support* between chunks of text, as required to construct bipolar argumentation frameworks (BAFs) (Cayrol and Lagasquie-Schiex 2005), and show how it can be deployed effectively also with small data sets. BAFs can be seen as graphs with arguments as nodes and two types of directed edges between nodes, representing attack and support between the arguments. An example of a BAF is given in Figure 1. Mining attack and support from natural language texts is the main task in **relation-based argument mining** (RbAM), which amounts to identifying arguments in text as well as dialectical relations between these arguments (Carstens and Toni 2015; Bosc, Cabrio, and Villata 2016; Menini et al. 2018).

We define a deep learning architecture based on a long–short term memory (LSTM) model (Hochreiter and Schmidhuber 1997) to determine relations of *attack*, *support*, and *neither attack nor support* between any two pieces of text. Within our deep network architecture, each input text is fed into a LSTM model, which produces a vector representation of the text being analyzed. The two vectors are then merged using various techniques and the resulting vector is finally fed into a softmax classifier, which predicts the label for the relation between the two texts. We achieve 89.53% accuracy using LSTMs and concatenation as the merge layer, considerably outperforming the results with feature-based supervised classifiers reported in the study that introduced the corpus used in this article (Carstens and Toni 2015, 2017).[1]

We then test our best-performing deep learning model on different data sets consisting of news article headlines to determine whether these *support* tweets, and show that our model generalizes well. We use two data sets introduced in Tan (2017): one consisting of pairs of tweets–headlines related to the FBI's investigative involvement in Hillary Clinton's e-mail leak and the second one adapted from Guo et al. (2013). For example, consider the following:

news headline: *"Stocks Push Higher."*

tweet: *"NYTimes: Markets Ride High as Small Investors Return."*

Our model can predict that the headline *supports* the tweet. Making these predictions can be a useful task in fact-checking settings, particularly for testing whether tweets are backed by any information. Indeed, the Fake News Challenge[2] indicates that determining agreement toward a statement is a useful step toward determining its truthfulness.

1 See https://www.doc.ic.ac.uk/~lc1310/.
2 http://www.fakenewschallenge.org/.

We then show that our LSTM model can be used to extract full BAFs (as opposed to singling out individual relations) from text (e.g., reviews). For example, consider the following two reviews about a hotel:

r_1: *"I recommend the hotel, it has nice rooms."*

r_2: *"The room was very dusty."*

The extracted BAF may be as shown in Figure 1. The BAFs extracted from reviews can be seen as arguments for evaluating the "goodness" of the item being reviewed and thus showing the reasons as to whether to recommend that product or service by providing an argumentation chain of other users' arguments.

Once a BAF has been extracted from text, we can use argumentative reasoning to evaluate the arguments in the BAF and in particular the dialectical "goodness" of an item being reviewed. We use a form of argumentative reasoning supported by the Discontinuity-Free Quantitative Argumentation Debate (DF-QuAD) algorithm (Rago et al. 2016) that computes the dialectical strength of arguments. For example, in the BAF in Figure 1, the dialectical strength of argument a is greater when having only the support from argument b compared with having the support from b, which is, in turn, attacked by argument c.

The dialectical strength of arguments can then be used to contribute new **argumentative features** for machine learning classifiers. Our new argumentative features capture the impact of each review on determining how "good" an item is with respect to all reviews about that item. Thus, our argumentative features can be seen as adding a semantic layer to the analysis of reviews as it uses information from discourse and the wider context represented by the other reviews about that item. We deploy these argumentative features to help detect deceptive reviews.

Detecting deceptive reviews is an important problem, studied, for example, in Crawford et al. (2015). It has an effect in e-commerce, as deceptive reviews may persuade potential customers to buy a company's product/service (if they are positive) or discourage customers from purchasing (if they are negative). Some reviews may be maliciously written by competitors in order to defame a company's products or to promote their own products/services. The state-of-the-art in this context is to extract features from reviews using standard syntactic analysis given by Natural Language Processing (NLP) when using machine learning techniques (Crawford et al. 2015). We experiment with the use of argumentative features with random forests (RFs) (Breiman 2001) in two domains (hotels and restaurants), using the data set from Ott, Cardie, and Hancock (2013) and Li et al. (2014). These are the gold standard in deception detection for reviews but are rather small (1,600 hotel reviews and 400 restaurant reviews).

We show experimentally that combining deep learning and argumentative reasoning outperforms standard supervised machine learning techniques in this setting, with improvements varying from 1% to 3% on the hotel data set when using a subset of the data set.

This article builds upon and extends preliminary work as follows. The argumentative features were introduced by Cocarascu and Toni (2016) and the deep learning architecture for RbAM is described in Cocarascu and Toni (2017). In the current article, we use the deep learning architecture to predict support from news headlines and tweets. Moreover, we extend the method for extracting BAFs from reviews using topic modeling and deep learning in Section 6 and report results in two domains, hotel (see

Table 8) and restaurant (see Table 9). We show that deep learning, combined with argumentative reasoning, improves on the task of determining whether a review is truthful or deceptive and is also able to handle the small data set issue. Albeit small, the improvements show promise in the integration of deep learning and symbolic, argumentative reasoning.

The remainder of this article is organized as follows. In Section 2, we discuss related work. In Section 3, we review relevant background information in LSTM models and argumentation and give an overview of the data sets used in this article. In Section 4, we describe our deep learning architecture. We report the performance of our deep learning model in identifying the support relation between headlines of news articles and tweets in Section 5. In Section 6, we describe our approach to extracting arguments from reviews and building BAFs, and define the argumentative features drawn from these frameworks. We also report results when using these argumentative features in determining whether reviews are deceptive. In Section 7, we show how deep learning and argumentative reasoning can handle the case of small data sets in our domain of interest. We conclude the article and propose directions for future work in Section 8.

2. Related Work

This work focuses on using deep learning combined with argumentative reasoning with frameworks obtained by RbAM for deception detection. In this section, we review related work in RbAM and argument mining in general and in detection of deception in reviews.

2.1 Argument Mining

Existing argument mining (AM) approaches focus on a variety of 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) (see Lippi and Torroni [2016] for an overview). Classification of pairs of sentences, amounting to identifying relations between texts, has recently received a great deal of attention. In particular, in this article we focus on the RbAM task as defined by Carstens and Toni (2015), which aims to automatically identify relations between arguments to create BAFs (Cayrol and Lagasquie-Schiex 2005). Carstens and Toni (2017) obtain 61.8% accuracy on a news articles corpus using support vector machines (SVMs) and features such as distance measures, word overlap, sentence metrics, and occurrences of sentiment words. Cabrio and Villata (2012, 2013) use textual entailment to identify arguments within text and to determine the relations between these arguments. Dusmanu, Cabrio, and Villata (2017) focus on the task of mining arguments from Twitter, distinguishing between opinions and factual arguments and identifying the source of these arguments using logistic regression (LR) and RFs. Some works focus on identifying supporting arguments in relevant documents given a claim (Hua and Wang 2017) and in reviews (Poddar, Hsu, and Lee 2017). Other works focus on different AM tasks than the ones we address in this article, such as identifying argument components, claims, and premises, and the links between these—for example, using LSTMs (Eger, Daxenberger, and Gurevych 2017; Niculae, Park, and Cardie 2017; Potash, Romanov, and Rumshisky 2017).

Several authors have used neural network models for tasks related to argument mining. In particular, Yin et al. (2016) propose three attention mechanisms for

convolutional neural networks to model pairs of sentences in tasks such as textual entailment and answer selection. Determining the relations in a Stanford natural language inference (SNLI) sentence pair is addressed by Bowman et al. (2015), using stacked LSTMs with the bottom layer taking as input the concatenation of the premise and of the hypothesis; and by Bowman et al. (2016), using TreeLSTM-like models with shared parameters between the premise and the hypothesis. Recognizing textual entailment between two sentences is also addressed by Rocktäschel et al. (2015), using LSTMs and word-by-word neural attention mechanisms on the SNLI data set. Liu et al. (2016) propose two models that capture the interdependencies between two parallel LSTMs encoding the two sentences for the tasks of recognizing textual entailment and matching questions and answers, respectively. A bidirectional recurrent neural network (BiRNN) with a word embedding-based attention model is used to determine whether a piece of evidence supports the claim of a support/attack relation using a data set of 1,000 pairs of sentences in Koreeda et al. (2016). In addition, Bosc, Cabrio, and Villata (2016) use a corpus consisting of tweets to determine *attack* and *support* relations between tweets. Using an encoder–decoder architecture with two LSTMs where the second LSTM is initialized with the last hidden state of the first LSTM, they obtain negative results (0.2 F_1 score for *support* and 0.16 F_1 score for *attack*). Further, Menini et al. (2018) identify *attack* and *support* relations in political speeches from the 1960 presidential campaign consisting of 1,462 pairs of arguments and achieve 72% accuracy using SVMs.

There are few studies in the AM community that use deep learning models to determine relations between arguments, but of a different kind than *attack* and *support* as in our work. Notably, Habernal and Gurevych (2016) experiment with both bidirectional LSTMs (BiLSTMs) and BiLSTMs extended with an attention mechanism and a convolution layer over the input to determine the class that explains why a certain argument is more convincing than the other in the pair. Whereas they focus on determining convincingness, we focus on identifying *attack*, *support*, or *neither* relations between arguments.

2.2 Review Spam Detection

Review spam detection has recently received a great deal of attention. An overview of the machine learning techniques and features used to detect review spam is given by Crawford et al. (2015). Much of the previous work on detecting deceptive reviews focus on detecting either reviews (e.g., opinion spam) (Ott et al. 2011; Shojaee et al. 2013; Fusilier et al. 2015) or deceptive spammers (Lim et al. 2010; Mukherjee, Liu, and Glance 2012). Other work focuses on detecting single review spammers (Lim et al. 2010) and group review spammers (Mukherjee, Liu, and Glance 2012). Sandulescu and Ester (2015) look at identifying reviews written by the same person but under different names. Given that the majority of users write a single review, others focus on identifying singleton deceptive reviews using, for example, multiscale 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 (Xie et al. 2012).

Different forms of machine learning have been used in the literature to detect deceptive behavior, notably unsupervised (Mukherjee et al. 2013), semi–supervised (Fusilier et al. 2015), and supervised (Ott et al. 2011; Ott, Cardie, and Hancock 2013; Shojaee et al. 2013; Li et al. 2014) 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 (Jindal and Liu 2007; Li et al. 2011; Rout et al. 2017). Some

previous work singles out quantity, specificity, diversity, non-immediacy, as well as task specific features such as affect, expressivity, complexity, uncertainty, and informality (Zhou et al. 2004; Fuller et al. 2006).

Hai et al. (2016) use review spam detection for different domains (hotel and restaurant) as a multitask learning problem by sharing the knowledge from training applied to each task and a graph regularizer for each model to incorporate unlabeled data. Mukherjee, Dutta, and Weikum (2017) use a model based on latent topic models in combination with limited metadata to compute a credibility score for reviews as well as to identify inconsistencies that appear between a review and the overall characterization of an item both for the item and for each latent facet. Viviani and Pasi (2017) proposed a multi-criteria decision-making strategy to identify fake reviews by evaluating the impact of each criterion on the veracity of reviews and using various methods to compute the overall veracity score. Ren and Ji (2017) proposed a three-stage system for detecting deceptive reviews: Learn sentence representations from word vectors, learn document representations from sentence vectors, and finally learn using the document vectors as features.

3. Background

Our work draws primarily on Recurrent Neural Networks and argumentative reasoning with Argumentation Frameworks. In this section, we elaborate on relevant background from the two fields as well as on the data sets used in this article.

3.1 Recurrent Neural Networks and Variations

Recurrent neural networks (RNNs) (Elman 1990; Mikolov et al. 2010) are a type of neural network in which some hidden layer is connected to itself so that the previous hidden state can be used along with the input at the current step. However, RNNs tend to suffer from the vanishing gradients problem (Bengio, Simard, and Frasconi 1994) while trying to capture long-term dependencies.

LSTM models (Hochreiter and Schmidhuber 1997) address this problem by introducing memory cells and gates into networks. LSTM models are a type of RNN that use memory cells to store contextual information and three types of gates (input, forget, and output gates) that determine what information needs to be added or removed in order to learn long-term dependencies within a sequence.

One problem with RNNs/LSTM models in NLP is that they do not make use of the information of future words. BiRNNs/BiLSTMs (Schuster and Paliwal 1997) solve this problem by using both previous and future words. This neural model processes the input sequence with two RNNs—one in the forward and one in the backward direction—resulting in two vectors for each input.

3.2 Argumentation Frameworks

(Abstract) argumentation frameworks (AAFs), introduced by Dung (1995), 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 $attacks \subseteq AR \times AR$. BAFs extend AAFs by considering two independent binary relations between arguments: attack and support (Cayrol and Lagasquie-Schiex 2005). Formally, a BAF is any $\langle AR, attacks,$

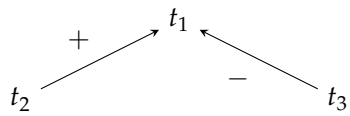
supports⟩ where *attacks* ⊆ $AR \times AR$ and *supports* ⊆ $AR \times AR$. For example, consider the following three texts:

t_1: *"We should grant politicians immunity from prosecution"*

t_2: *"Giving politicians immunity from prosecution allows them to focus on performing their duties"*

t_3: *"The ability to prosecute politicians is the ultimate protection against the abuse of power"*

Here t_2 *supports* t_1, t_3 *attacks* t_1, and t_2 and t_3 *neither attack nor support each other*. The BAF is:



Semantics of BAFs can be defined in terms of a notion of **strength** (e.g., Aurisicchio et al. 2015), namely, a function from AR to (a suitable subset of) real numbers. As in Aurisicchio et al. (2015), 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.

Many different notions of strength have been proposed in the literature, mostly with very similar properties (see, e.g., Rago, Toni, and Baroni 2018). In this article, we use the DF-QuAD algorithm (Rago et al. 2016). This is defined for restricted types of BAFs that can be represented as trees. In DF-QuAD, arguments are equipped with a base score that amounts to an intrinsic (non-dialectical) strength of arguments. This strength is then altered to give the final (dialectical) strength based on combining the (dialectical) strength of attacking and supporting arguments. The resulting strength of arguments is determined by aggregating the strength of attackers against and supporters for these arguments. The strength aggregation function $\mathcal{F}$, given n arguments with strengths $v_1, ..., v_n$, is defined as follows:

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

The 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 attackers) 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 $v_s = \mathcal{F}(v'_1, ..., v'_m)$:

$$\mathcal{C}(v_0, v_a, v_s) = \begin{cases} v_0 & v_a = v_s \\ v_0 - v_0 \cdot |v_s - v_a| & v_a > v_s \\ v_0 + (1 - v_0) \cdot |v_s - v_a| & v_a < v_s \end{cases}$$

Finally, for any argument $a \in AR$ with base score v_0 and n attackers with strengths $v_1, \ldots, v_n$ and m supporters with strengths $v'_1, \ldots, v'_m$ the strength of argument a is defined as follows:

$$strength(a) = \mathcal{C}(v_0, \mathcal{F}(v_1, ..., v_n), \mathcal{F}(v'_1, ..., v'_m))$$

3.3 Relational Data Set

Determining relations between any texts can be viewed as a three-class problem, with classification labels $L = \{attack, support, neither\}$. We use a data set[3] adapted from the one used in Carstens and Toni (2017), covering topics such as UKIP and opinions about movies, technology, and politics, where *attack* relations represent 31% of the data set, *support* relations represents 32% of the data set, and *neither* relations represent 37% of the data set.

We have also explored the use of other corpora (such as the AIFdb corpus,[4] which has a finer-grained analysis of argumentative types, and SNLI (Bowman et al. 2015), used in recognizing textual entailment, contradiction, and neutral relations), which we ultimately decided not to include because of their structure not being amenable to our analysis, for the reasons we give in the following.

The AIFdb corpus consists of graphs with two types of nodes: information nodes (I-nodes) and scheme nodes (S-nodes). S-nodes represent relations between I-nodes and may in turn be of different kinds. These kinds are rule application nodes (RA-nodes), representing inference rules, and conflict application nodes (CA-nodes), representing generic conflicts. Further, transition applications nodes (TA-nodes) are special kinds of S-nodes connecting locution nodes (L-nodes, special type of I-nodes) to capture dialogue flow. Although we initially hypothesized that CA-nodes could indicate attack and RA- and TA-nodes support for RbAM, we found no evidence in practice that this is the case. For example, a TA relation between *"No parent in the family is in work"* and *"We have a huge problem with unemployment"* does not indicate a clear *support* relation, in the sense of RbAM.

The SNLI corpus contains 570k sentence pairs labeled as *entailment*, *contradiction*, or *neutral*. These relations may seem to have some similarity with the relations of interest to RbAM, namely *support*, *attack*, or *neither (support nor attack)*, respectively. However, the type of sentence pairs found in this corpus is different from the types of texts we are interested in analyzing in RbAM. To illustrate, an example of entailment pair in the SNLI corpus is as follows: *"A soccer game with multiple males playing"* and *"Some men are playing a sport."* We are instead interested in dialectical relations (e.g., of support), as between the following two texts: *"I believe that what UKIP is doing is vital for this country"* and *"It is because of UKIP that we are finally discussing the European question and about immigration and thank goodness for that."*

3.4 Reviews Data Sets

The gold standard for deceptive reviews consists of positive and negative hotel reviews of 20 Chicago hotels (Ott, Cardie, and Hancock 2013), 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 (Li et al. 2014). Existing studies have focused on detecting deceptive hotel reviews (Ott et al. 2011), identifying positive and negative deceptive hotel reviews (Ott, Cardie, and Hancock 2013) and cross-domain deception on the more recent data set (Li et al. 2014).

3 https://www.doc.ic.ac.uk/~oc511/ACMToIT2017_dataset.xlsx.
4 http://corpora.aifdb.org/.

The hotel data set that we use consists of 1,600 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 Web sites, and 400 deceptive positive reviews and 400 deceptive negative reviews from Turkers (Ott, Cardie, and Hancock 2013). The restaurant data set that we use consists of 400 reviews about 10 restaurants, 200 deceptive reviews, and 200 truthful reviews (Li et al. 2014).

4. Deep Learning for RbAM

We propose a deep learning architecture to capture argumentative relations of *attack*, *support*, or *neither support nor attack* between any two pieces of text using LSTM networks. In RbAM, we assume that if one sentence attacks/supports another sentence, then both are considered to be argumentative, irrespective of their standalone argumentativeness.

4.1 Architecture

Several types of deep learning architectures have been used in AM or similar tasks where sentence pairs need to be classified. These include LSTMs (Bowman et al. 2015, 2016; Liu et al. 2016); encoder–decoder LSTMs (Bosc, Cabrio, and Villata 2016), attentional LSTMs (Rocktäschel et al. 2015; Koreeda et al. 2016; Liu et al. 2016), which use a soft attention mechanism so that the representation of one piece of text depends on the representation of the other piece of text; and (attention-based) convolutional neural networks (Habernal and Gurevych 2016; Yin et al. 2016).

LSTMs can be used to encode each text separately and then merged in order to classify the argumentative relation. LSTMs have been proven successful in learning sentence representations in AM or similar tasks. We experimented with both LSTMs and BiLSTMs to determine the type of relation—*attack*, *support*, *neither attack nor support*—between two texts. We do not impose texts to be single sentences, but we do however limit the input sequences to 50 words. We pad the inputs with size smaller than this threshold with zeros at the end to produce sequences of exactly 50 words. We initialize the word embeddings for our deep learning architecture with the 100-dimensional GloVe vectors (Pennington, Socher, and Manning 2014).[5] The words that do not appear in the vectors are treated as unknown.

We use two parallel (Bi)LSTMs to model the two texts. Indeed, based on our assumption that if one sentence attacks/supports another sentence, then both may be considered to be argumentative, irrespective of their standalone argumentativeness, we opted for two classifiers to model the two texts independently of one another, and then merge the results.

Each (Bi)LSTM model produces a vector representation of the text being analyzed without any context from the other text. In the experiments we set the dimension of the word embedding to be 100 and the LSTM dimension to be 32. We have experimented with both *ReLU* and *sigmoid* activation functions for the two LSTMs in our model (see Section 4.2); although ReLU is not commonly used for LSTMs, it gives rise to good experimental results in our model. We merge the two vectors obtained from the (Bi)LSTMs using various approaches and feed the resulting vector to a softmax

5 Pennington, Socher, and Manning (2014) computed the 100-dimensional GloVe embeddings on a a dump of English Wikipedia pages from 2014 consisting of 400k words.

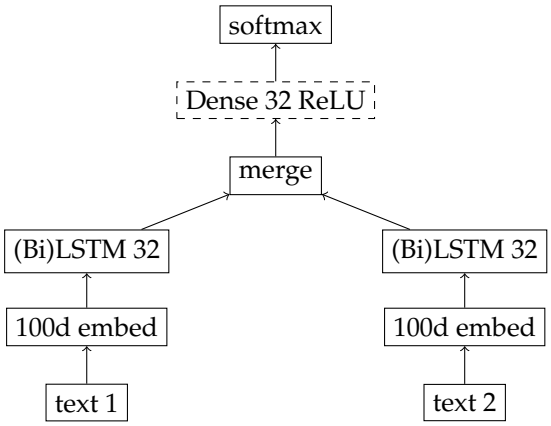


Figure 2
Our classification architecture. Two LSTMs are run with one text each. We tested two types of LSTMs: unidirectional and bidirectional. The dashed layer (Dense 32 ReLU) is optional and its inclusion in the architecture and impact on performance was tested in our experiments.

classifier, which predicted the label from $L = \{attack, support, neither\}$ for the relation between the two texts. We have experimented with two types of merge layer: *sum*, which performs elementwise sum, and *concat*, which performs tensor concatenation. After the merge layer, we also add a dense feedforward layer to test whether its inclusion has an impact on the results. In particular, we conducted experiments to test the influence of a dense feedforward layer before softmax to determine the type of class from L. We run experiments using the same architecture with BiLSTMs as well as (unidirectional) LSTMs. Figure 2 describes the architecture that we use for determining the type of relation from L between texts.

4.2 RbAM Results

We experimented with unidirectional LSTMs and BiLSTMs. In both cases we set the LSTM dimension to 32 (see Section 4.1), as this proved to be the best among alternatives we tried (64, 100, 128). We trained for 50 epochs or until the performance on the development set stopped improving (thus effectively avoiding overfitting by early stopping), using a mini-batch size of 128 and cross-entropy loss. To avoid overfitting, we applied dropout on each LSTM before the merge layer with probability 0.2. We did not use dropout on the recurrent units. The model parameters were optimized using the Adam method (Kingma and Ba 2014) with learning rate 0.001. Indeed, this method provided better performance than alternative optimizers we tried (Adagrad, Adadelta, and RMSprop). We run the same experiments (with the same hyperparameters) for unidirectional LSTMs and BiLSTMs. The values for the hyperparameters are shown in Table 1.

As a baseline we used LR and unigrams obtained from concatenating the input pairs of texts. LR proved to give the best results among the alternatives considered, SVMs and RFs, which typically perform well and were also used in Carstens and Toni (2015, 2017), which introduced the corpus used in this article. We did not use the results reported in Carstens and Toni (2017) as baselines because these results are obtained for data sets from various sources, including but not limited to the data we have used.

Table 1
Hyperparameters for our LSTM and BiLSTM models.

Hyperparameter	Value
Dropout	0.2
Embedding size	100
Maximum sequence length	50
LSTM size	32
Dense size	32
Batch size	128

The results on each data set coming from a different source are not reported in Carstens and Toni (2017).

We run 10 stratified fold cross-validations (so that each fold is a good representative of the whole) for 5 times, choosing ReLU and sigmoid as alternative activation functions for the (Bi)LSTMs. Networks with ReLU have a lower run time and tend to show better convergence (Talathi and Vartak 2015). However, ReLU has the disadvantage of dying cells (the dying ReLU problem), but this can be overcome by using a variant called Leaky ReLU. We report experiments with using both the pre-trained word representations (freezing the weights during learning) as well as learning the weights during training (trained embeddings). We experimented with three types of (unidirectional) LSTM models, one with a *sum* merge layer and two with the vectors from LSTMs being *concatenated*. In the case of using a *concatenation* layer, we explored whether having a feedforward layer after the merge layer results in any differences in performance. We report only results for BiLSTMs using concatenation and the feedforward layer, as this was the best performing combination for BiLSTMs. We provide in Tables 2 and 3, with ReLU and sigmoid activation functions, respectively, the results obtained using both BiLSTMs and (unidirectional) LSTMs with the two types of merge layers.

With ReLU activation functions (Table 2), we achieved 89.53% accuracy and 89.07% F_1 by concatenating the output of the two separate LSTMs. Unexpectedly, BiLSTMs performed worse than LSTMs (Table 2 only includes the best performing BiLSTM instance of the architecture, using concatenation and the feedforward layer). We believe this is because of the size of the data set and that this effect could be diminished by acquiring

Table 2
5×10 fold cross-validation results, using *c(oncat)* or *s(um)* for merging the output of the two (Bi)LSTMs, with (non-)trained embeddings; T(*True*)/F(*False*) represent inclusion/omission, respectively, of the Dense 32 ReLU layer. *std* represents standard deviation of 5×10 fold cross-validation.

Baseline	**A**%	**P**%	**R**%	**F_1**%
LR (unigrams)	77.87	78.02	77.87	77.89

Model/	**Non–trained embeddings**				**Trained embeddings**					
Merge/Dense	**A**%	**P**%	**R**%	**F_1**%	**A**%	**P**%	**R**%	**F_1**%	**Astd**	**F_1**std
BiLSTM/c/T	60.72	64.36	52.64	57.36	70.66	73.18	62.96	66.93	2.06	4.60
LSTM/c/F	68.25	72.39	59.07	64.38	89.53	90.80	87.67	89.07	0.47	0.73
LSTM/c/T	68.68	72.77	58.21	63.49	90.02	90.89	88.26	89.41	2.09	2.92
LSTM/s/T	64.21	69.18	51.07	57.09	84.84	86.75	79.98	82.35	5.02	9.26

Table 3
5 × 10 fold cross-validation results, using *c(oncat)* or *s(um)* for merging the output of the two (Bi)LSTMs, with (non-)trained embeddings; T(*True*)/F(*False*) represent inclusion/omission, respectively, of the Dense 32 sigmoid layer. *std* represents standard deviation of 5 × 10 fold cross-validation.

Baseline	**A%**	**P%**	**R%**	**F_1%**						
LR (unigrams)	77.87	78.02	77.87	77.89						
Model/	**Non–trained embeddings**				**Trained embeddings**					
Merge/Dense	**A%**	**P%**	**R%**	**F_1%**	**A%**	**P%**	**R%**	**F_1%**	**Astd**	**F_1std**
BiLSTM/c/T	71.9	74.12	71.9	72.44	93.42	93.74	93.42	93.4	0.36	0.37
LSTM/c/F	59.02	60.86	59.02	56.48	91.4	91.58	91.4	91.38	0.38	0.36
LSTM/c/T	43.84	34.4	43.84	32.2	75.12	70	75.12	70	7.57	9.94
LSTM/s/T	46.52	41.36	46.52	37.12	72.66	68.04	72.66	66.94	5.70	7.49

more data. For the LSTM model with trained embeddings, the accuracy varied between 84.84% and 90.02%. Concatenating the LSTMs' output vectors yields better performance than performing element-wise sum of the vectors. One reason would be that this allows the system to encode more features, allowing the network to use more information. Using the default, pre-trained word embeddings and freezing the weights during learning yields worse results compared to the baseline. This can be attributed to the fact that the quality of word embeddings is dependent on the training corpora. Training the word embeddings results in better performance compared with the baseline with improvements of up to 12% in accuracy and up to 11.5% in F_1. In all cases, training the word embeddings results in dramatic improvements compared to freezing the embedding weights during learning, varying from 9.9% to 21.3% increase in accuracy and up to 25% in F_1. We also report the standard deviation of our models with trained embeddings. This shows that our best model (LSTMs with a concatenation layer) is stable and performs consistently on the task considered. Using one-way ANOVA, the result is significant at $p < 0.05$ (the F-ratio value is 145.45159, the p-value is < 0.00001).

With sigmoid activation functions (Table 3), we achieved 91.4% accuracy and 91.38% F_1 by concatenating the output of the two separate LSTMs. We have chosen this as our best model because the std of F_1 was the smallest. For the LSTM model with trained embeddings, the accuracy varied between 72.66% and 93.42%. Again, concatenating the LSTMs' output vectors yields better performance than performing element-wise sum of the vectors. Using the default, pre-trained word embeddings and freezing the weights during learning yields worse results compared with the baseline. Training the word embeddings results in better performance compared with the baseline, with improvements of up to 15.5% in accuracy and up to 15.51% in F_1. In all cases, training the word embeddings results in dramatic improvements compared with freezing the embedding weights during learning, varying from 21.5% to 32% increase in accuracy and up to 38% in F_1. We also report the standard deviation of our models with trained embeddings. This shows that our best model (LSTMs with a concatenation layer) is stable and performs consistently on the task considered. Using one-way ANOVA, the result is significant at $p < 0.05$ (the F-ratio value is 4.23093, the p-value is < 0.011629).

5. Identifying Whether News Headlines Support Tweets

We test whether the system described in Section 4 performs well on different types of texts: tweets and news headlines. More specifically, we are interested in determining

Table 4
Examples from the tweet data sets.

Data Set	Tweet	Headline
Tan 2017	Crooked Hillary Clinton who I would love to call Lyin' Hillary is getting ready to totally misrepresent my foreign policy positions	Hillary Clinton Calls Donald Trump's Foreign Policy Ideas Dangerously Incoherent
Tan 2017	Hillary's Two Official Favors To Morocco Resulted In $28 Million For Clinton Foundation	New Clinton Foundation donation policy sparks fresh criticism
Adapted from Guo et al. 2013	NYTimes: Markets Ride High as Small Investors Return	Stocks Push Higher
Adapted from Guo et al. 2013	Bloomberg has donated over $1 billion to Johns Hopkins	At $1.1 Billion Bloomberg Is Top University Donor in U.S.

Table 5
Performance of our model on the tweet data sets.

Data Set	P%	R%	F_1%	Number of examples
Tan 2017	0.59	0.97	0.73	30
Adapted from Guo et al. 2013	0.97	0.90	0.94	840

whether our proposed model correctly identifies that a headline of a news article *supports* a tweet. We use the two data sets introduced in Tan (2017):[6] one consisting of pairs of tweets–headlines related to the FBI's investigative involvement in Hillary Clinton's e-mail leak, with 30 support relations; and the second one adapted from Guo et al. (2013), with 840 support relations. The latter originally had tweets that explicitly contained URLs to CNN or the *New York Times*. The authors extracted the news titles from the URLs with the aim of finding the most relevant article to the tweet.

We used the filtered data set as in Tan (2017), discarding headlines such as "Christmas Where I live," "how to get rid of old gadgets," and question type headlines, such as "Will the big four become two?" Some examples from the data set can be seen in Table 4.

The performance of our model described in Section 4 on the two tweet data sets is given in Table 5. On the tweet data sets consisting of 30 and 840 examples, our model yields 73% F_1 and 94% F_1, respectively; thus it generalizes well.

Identifying news headlines that support tweets is useful in fact-checking settings, particularly in testing whether tweets are backed by any information. Indeed, the Fake News Challenge[7] indicates that determining agreement toward a statement is a useful step toward determining its truthfulness.

6 https://www.doc.ic.ac.uk/~oc511/data.json.
7 http://www.fakenewschallenge.org/.

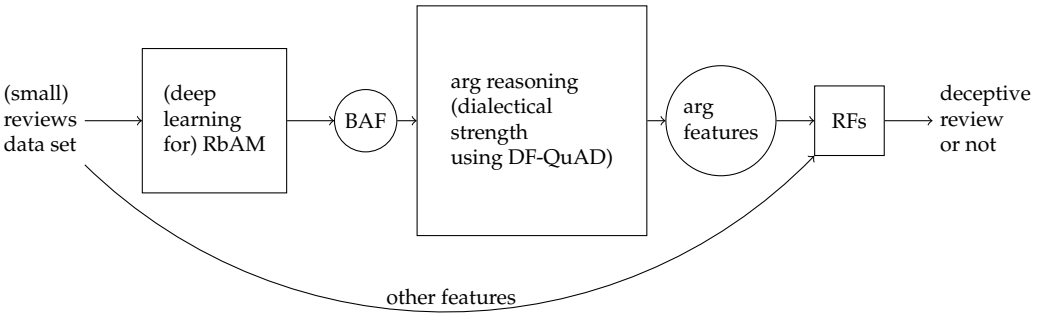


Figure 3
Overview of how (deep learning for) RbAM and argumentative (arg) reasoning and features are used for detecting deceptive reviews.

6. Mining Bipolar Argumentation Frameworks for Detecting Deceptive Reviews

Our approach to detecting deceptive reviews is based on mining BAFs constructed from arguments that are clustered based on the topics extracted from reviews. We explore different approaches for identifying topics in reviews, ranging from associating each noun encountered in reviews with a topic, to more advanced techniques related to topic modeling, such as latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) and non-negative matrix factorization (NMF) (Lee and Seung 1999). We compare two methods for RbAM used for constructing the BAFs: a supervised classifier that uses syntactic and semantic features, and the deep learning architecture based on LSTMs explained in Section 4.1. The BAFs extracted from the reviews will serve to provide new argumentative features, which are then used, along with other features, to determine whether a review is deceptive or not. We show that combining deep learning and argumentative reasoning gives better performance than standard machine learning techniques for deception detection.

An overview of how (deep learning for) RbAM and argumentative reasoning and features are used for detecting deceptive reviews in our system is given in Figure 3. The deep learning model identifies arguments and relations of *attack*, *support*, and *neither attack nor support* between arguments from a set of reviews. Using the attack and support relations extracted from the reviews, we construct BAFs and compute the dialectical strength of arguments in these BAFs using DF-QuAD. This contributes new argumentative features which, along with other syntactic features previously identified in studies of deception, are fed into RF to determine whether a review is truthful or deceptive.

6.1 Building a Topic-Dependent BAF

The procedure for constructing a topic-dependent BAF is described in detail here:

1. Split each review into sentences (where each sentence is a potential argument).
2. Identify topics in reviews and the sentences (potential arguments) related to each topic.

3. For each topic, run the RbAM classifier on the sentences associated with this topic to determine the relations between them.
4. Construct the BAF.

This process is done for the set of reviews associated with each *item* in the review (i.e., the hotel data set contains reviews for 20 hotels; we run this process for each hotel). We assume that each argument extracted from the reviews is contained in a sentence. Thus, each review is mapped to one or more such arguments.

6.1.1 Step 1. The first step amounts to identifying parts of the reviews that form the BAF's arguments. Identifying arguments in text is a complex task in general, and may require identifying components and boundaries of arguments (Lippi and Torroni 2016). In this article, we simply opt for equating sentences and potential arguments. Concretely, we split each review into sentences with a pre-trained tokenizer for English from *nltk* (Bird, Klein, and Loper 2009). Sentences containing *but*, *although*, *though*, *otherwise*, *however*, *unless*, or *whereas* are split because 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). The sentiment polarity of each sentence is determined using sentiment analysis from the *pattern.en* module (De Smedt and Daelemans 2012), which uses a lexicon of frequently used adjectives in product reviews annotated with scores for sentiment polarity.

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* (−)[8]

6.1.2 Step 2. The second step amounts to grouping potential arguments resulting from the first step by topic, to facilitate identifying (and render more meaningful) relations of attack and support between these arguments in the third step. Topic extraction can be performed in many alternative ways, including associating the lemmatized nouns from the reviews to topics (e.g., given the example reviews, topics may be *room*, *food*, *service*, *Wi-Fi*), LDA, and NMF. We choose to deploy LDA and NMF, as these are able to better uncover the underlying semantic structure of a set of documents by identifying (words that belong to) topics. LDA is a generative probabilistic model for discrete data.

8 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."*

We use LDA with online variational Bayes algorithm. Each document is represented as a bag of words, where the vocabulary represents the set of words in the data set. Being a probabilistic graphical model, we only need term count features for LDA. NMF finds two non-negative matrices (W, H), whose product approximates the non-negative matrix X representing the corpus of documents. Given N documents and M words in the vocabulary, $X = W \cdot H$ where the number of topics $K \ll N, M$, $X \in \mathbb{R}^{N \times M}$, $W \in \mathbb{R}^{N \times K}$, and $H \in \mathbb{R}^{K \times M}$. We compute the term frequency–inverse document frequency (*tf-idf*) and we use NMF with 0.1 regularization to optimize the squared Frobenius norm.

For both LDA and NMF, we remove stop words and ignore the terms that appear in only one document or in at least 95% of the documents. In the experiments described in Section 6.3, we identify 35 topics using LDA and NMF and select the top 25 words for each topic. We have tried different values for the number of topics, as well, subjectively analyzing some of the topics extracted to see if they are coherent. We have opted for the number of topics and words per topic that gave the best performances in the experiments. Other techniques could be used in order to select model parameters (e.g., Zhao et al. 2015).

After having identified topics, we identify the sentences/arguments related to these topics. In the case of topics being associated with nouns, we extract the sentences that contain that specific topic/noun. For LDA/NMF, we extract the sentences that contain any of the top words associated with the topics extracted by these methods.

6.1.3 Step 3. The third step amounts to determining dialectical relations between any pair of sentences/arguments associated with the same topics. This can be viewed as a RbAM three-class problem, with classification labels $L = \{attack, support, neither\}$. In order to limit the number of comparisons, for each topic, we assume that a newer argument (with respect to time) can either *support*, *attack*, or *neither support nor attack* a previous argument, but not vice versa.

For comparison with our deep learning method introduced in Section 4.1, we have considered several methods for RbAM explored in the literature, including SVMs as in Carstens and Toni (2017) and LR and RFs as in Dusmanu, Cabrio, and Villata (2017). We opted for RFs with syntactic and semantic features, as this method was the best performing among the alternatives we tried.

The classification model based on RFs uses the features shown in Table 6. 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.

For our deep learning architecture, in order to identify the relations between arguments associated with each topic, we chose the model that yielded the best results as reported in Section 4.2: two parallel LSTMs with trained embeddings and a *concatenation* layer to merge the output of the two separate LSTMs.

6.1.4 Step 4. The fourth and final step amounts to constructing topic-dependent BAFs from reviews, for the purpose of assessing how "good" the item being reviewed is by assessing the dialectical strength of a special argument G (for "good") in the constructed BAFs. In turn, each topic t identified at Step 2 gives a special argument G_t (for "good as far as t is concerned") supporting G. Thus, intuitively, the stronger the various G_t in the computed BAFs, the stronger the G. In addition to these special arguments G_t supporting G, the BAFs also include relations between arguments related to topic t drawn from

Table 6
Overview of features used in determining relations between pairs of sentences.

Feature	Detail
Number of words	For each sentence
Average 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*)

reviews and G_t, so 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 nor attack another argument related to t from the same or some other review, as determined by RbAM at Step 3, then this argument a_t will either support or attack G_t, according to its polarity as determined by sentiment analysis.

For example, given reviews r_1 and r_2 from Section 6.1.1 and using nouns from reviews as topics as in Section 6.1.2, we obtain the BAF $\langle AR, attacks, supports \rangle$ with:

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

shown graphically in Figure 4 (where edges labeled $-$ represent attacks and edges labeled + represent supports).

We have imposed that arguments from more recent reviews can attack or support only arguments from less recent reviews or the special G_t arguments, rather than any arguments, independent of the order in which the reviews arose. We believe that this is legitimate, as it mimicks what humans experience when they write a review. In this case, they have full access to all previous reviews, thus being able to agree, disagree, or neither agree nor disagree with these reviews. This choice is also practical, allowing the limiting of comparisons performed by RbAM. It would be interesting to experiment with BAFs obtained from reviews without imposing the temporal restriction over comparisons, to check in particular whether the resulting BAFs could provide more effective argumentative features.

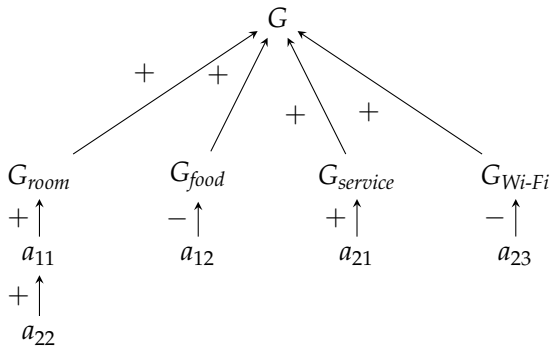


Figure 4
Topic-dependent BAF obtained from r_1,r_2 in Section 6.1.1.

6.2 From BAFs to Argumentative Features

In order to detect deceptive reviews, in addition to standard features used in previous studies, we associate argumentative features with each review, representing the impact of the review on how "good" an item (e.g., hotel or restaurant) is with respect to all reviews about that item. These new features are obtained from measuring the strength of arguments in the BAF built from all reviews related to the chosen item and in the BAF built from all reviews for that item except the one whose impact we aim at determining.

The BAFs obtained from sets of reviews, as described in Section 6.1, are, by construction, guaranteed to be in the restricted form of sets of trees. Note that these trees may have any (finite) breadth when choosing topics based on the nouns identified in the reviews or, in our specific set-up, breadth 35 when determining topics using LDA/NMF, and any depth as determined by the relations between arguments extracted from reviews.

Given that the BAFs are (sets of) trees, the strengths of arguments in these BAFs can be efficiently calculated recursively in terms of a strength aggregation function $\mathcal{F}$ and a combination function $\mathcal{C}$ as defined in Section 3.2. We then compute the strengths of arguments in the BAF built from all reviews for that item except the one whose impact we aim at determining.

Other methods for the calculation of strength are also deployable in practice, such as the game-theoretic approach of Baroni et al. (2017). We have, however, found that the DF-QuAD method can efficiently scale to support our experiments (Cocarascu and Toni 2016). Note that each different method for computing strength could conceptually be used to provide a new argumentative feature, in addition to the specific one using DF-QuAD that we use in this article.

For illustration, consider the BAF extracted earlier from reviews r_1, r_2 (see Figure 4). Assume a base score of 0.5 for all $a \in AR$ (we will use this same base score for all arguments in our experiments). The impact of review r is then given by the difference between the measure of how "good" the hotel/restaurant is deemed to be given all reviews R and how "good" 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 BAF all arguments from r_1, giving the BAF shown in Figure 5. 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.

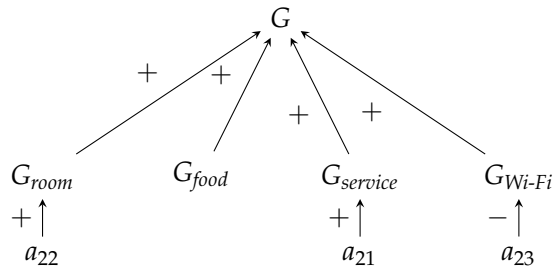


Figure 5
BAF obtained from removing (arguments from) r_1.

6.3 Detecting Deceptive Reviews: Experimental Results

We report the classification results on the task of determining whether a review is truthful or false on two domains, hotel and restaurant. We evaluate the performance of various techniques of extracting topics from reviews as presented in Section 6.1.2 and the impact our novel argumentative features have on the classifier's performance. All the results are obtained using 5-fold cross-validation and an ensemble method, RFs (Breiman 2001), with 10 trees in the forest, Gini impurity criterion, and the minimum number of samples required to split an internal node set to 2.

As a baseline, we extract features used previously in studies of deception (see Section 2.2). These features are the result of part-of-speech (POS) tag analysis using *nltk* and are summarized in Table 7.

Additionally, we include *tf-idf* features obtained from all reviews using *scikit–learn* (Pedregosa et al. 2011). To calculate these, we use the lemmas obtained by analyzing the lowercase form of words and their POS tag.

We present results of different approaches of constructing BAFs from reviews and hence including the argumentative features related to the impact each review has on the "goodness" of the item (hotel or restaurant) being reviewed (see Section 6.2 on how these features are computed). We experimented with two techniques for topic modeling, LDA and NMF (hence having features representing the impact of each review on the "goodness" of the item being reviewed for each of these methods, respectively).

Table 7
Features and the associated category.

Category	Features
Personalization	Number of self references
	Number of 2nd person pronouns
	Number of other references
	Number of group pronouns
Quantity	Number of sentences
	Number of words
	Number of nouns
	Number of verbs
Complexity	Average sentence length
	Average word length
Diversity	Lexical
Uncertainty	Number of modal verbs
	Number of modifiers

For each approach of topic modeling, we identify the topic that has the highest probability of being associated with the review, as well as all the topics with probability greater than 0.2 of being associated with the review. In both cases, we extract the sentences that contain any of the top words related to the topics that have been associated with the review. To identify the relations between arguments associated with each topic, we chose the best performing instance of our deep neural architecture trained on the full RbAM data set. We also report results when using the topic–noun approach and a RF classifier with features shown in Table 6. For each method of constructing the BAF, we create a new argumentative feature from computing the difference between the strength of arguments from all reviews and the strength of the arguments from all reviews except the one whose impact we aim at determining.

The classifiers' performances on the hotel data set are shown in Table 8. We see that adding the *tf-idf* features gives 76% accuracy, resulting in a dramatic improvement of 12% compared with the baseline, where the syntactic features from Table 7 were used. Using argumentative features extracted from the BAF constructed from topics being associated with nouns in reviews and using RFs for RbAM yields lower results compared to using syntactic features and *tf-idf* features, achieving 74.88% accuracy. Using argumentative features extracted from the BAF constructed using LDA and NMF for topic modeling and LSTMs for RbAM yields better results than using a standard classifier (RFs) and a simple topic extraction method (nouns $\sim$ topics) with accuracy 76.38%. Indeed, the best results are obtained using more advanced techniques for topic modeling rather than simple associations of topics $\sim$ nouns, and LSTMs for RbAM, with 0.38% improvement compared with using syntactic features and *tf-idf* features.

The classifiers' performance on the restaurant data set are shown in Table 9. Here as well, we see that adding the *tf-idf* features results in a dramatic improvement of 9% compared with the baseline. In contrast to the hotel data set, using argumentative features extracted from the BAF constructed from topics being associated with nouns in reviews and using RFs for RbAM results in an improvement of 1.5% compared with using only syntactic features. Using argumentative features extracted from the BAF constructed using LDA and NMF for topic modeling and LSTMs for RbAM also gives better results than using a standard classifier (RFs) and a simple topic extraction method (nouns $\sim$ topics) with accuracy 72.5%. Here again, the best results are obtained using more advanced techniques for topic modeling and LSTMs for RbAM, with 2.75% improvement compared with using syntactic features and *tf-idf* features.

We showed that combining deep learning and argumentative reasoning outperforms standard machine learning techniques for deception detection in both domains, hotel and restaurant. The results are encouraging and show that argumentative reasoning can indeed be used to improve classifications.

Table 8
Classifier performance on the hotel data set.

Standard features	Unigrams	Argumentative features	Topic model	RFs	
				Accuracy	F_1
✓	✗	✗	✗	63.81	63.6
✓	✓	✗	✗	76	75.83
✓	✓	✓	Nouns + RFs	74.88	74.71
✓	✓	✓	LDA & NMF + LSTM	76.38	76.18
✗	✗	✓	LDA & NMF + LSTM	50.0	33.33

Table 9
Classifier performance on the restaurant data set.

Standard features	Unigrams	Argumentative features	Topic model	RFs	
				Accuracy	F_1
✓	✗	✗	✗	60.75	60.6
✓	✓	✗	✗	69.75	69.69
✓	✓	✓	Nouns + RFs	71.25	71.15
✓	✓	✓	LDA & NMF + LSTM	72.5	72.4
✗	✗	✓	LDA & NMF + LSTM	50.0	33.33

We did not carry out any direct comparison with the results documented in the papers that introduced the reviews data sets we used (Ott et al. 2011; Ott, Cardie, and Hancock 2013; Li et al. 2014), as the tasks we focused on were different from the ones in the original papers. Concretely, Ott et al. (2011) experiment with a subset of the hotel data set used in this article, whereas Ott, Cardie, and Hancock (2013) focus on classifier performances on the hotel reviews data set based on the sentiment of the reviews (i.e., positive deceptive opinions and negative deceptive opinions). Furthermore, Li et al. (2014) focus on classifier performances in cross-domain adaptation and on performances on intra-domain multiclass classification tasks, with the aim of classifying reviews based on their source (reviews written by customers, employees, Turkers).

7. Deep Learning and Argumentation on Small Data Sets

The data sets described in Section 3.4 and used in this article differ in size: The hotel data set contains 1,600 reviews and the restaurant data set contains 400 reviews. We obtained improvements compared with the baseline as presented in Section 6.3 when using more advanced topic modeling techniques and LSTMs to determine relations between arguments extracted from the reviews.

The data sets used are small. Moreover, the size of the restaurant data set represents a quarter of the size of the hotel data set. We are interested to see whether, using a subset of the hotel data set, we can still achieve comparable or better results compared to using the entire data set to further explore the suitability of our methodology of combining deep learning for RbAM and argumentative reasoning by means of BAFs to cope with small data sets.

The results shown in Table 10 are obtained using 5-fold cross-validation. In all experiments we use LDA (as it has been shown that it learns more coherent topics than NMF [Stevens et al. 2012]) to extract topics from reviews and the best performing

Table 10
Classifier performance on the hotel data set using subsets of the data set.

Total number of reviews	RFs	
	Accuracy	F_1
1,600	76.69	76.48
1,200	79.83	79.74
800	79.63	79.48
400	77.75	77.62

instance of our deep neural architecture trained on the full RbAM data set. We see that, using a subset of the hotel data set, we achieve better results combining deep learning with argumentative reasoning compared with using the entire data set. We obtain 1% improvement in accuracy when using a quarter of the data set, and 3% accuracy improvement when using three quarters or half of the hotel data set. Using one-way ANOVA, the result is significant at $p < 0.05$ (the F-ratio value is 114.71494, the p-value is < 0.00001).

8. Conclusion and Future Work

We described a deep learning model for RbAM and used it in two settings: to determine whether news headlines support tweets and to detect deceptive reviews. Our deep learning architecture is based on LSTM networks to capture the argumentative relation of *attack*, *support*, or *neither attack nor support* between any two texts. We achieved 89.53% accuracy on the news articles data set of Carstens and Toni (2015). The results indicate that LSTMs may be better suited for this task than standard classifiers, as LSTMs are better at capturing long-term dependencies between words as they operate over sequences, which is the case for text.

We used our deep learning model on different data sets consisting of news article headlines that *support* tweets and showed that our model generalizes well. This suggests our model can be used for fact-checking by identifying information that supports tweets. Indeed, the Fake News Challenge indicates that determining agreement toward a statement is a useful step toward determining its truthfulness.

We also described a hybrid system combining deep learning and symbolic, argumentative reasoning to evaluate deception of online opinions and reviews. In addition to standard NLP features, we introduced argumentative features that capture semantic information from reviews represented as bipolar argumentation frameworks (BAFs). We show experimentally, for reviews about hotels and restaurants, that including the argumentative features yields better results in classifier performance, with improvement up to 0.38 percentage point for the hotel data set and an improvement of 2.75 percentage points for the restaurant data set.

Our experiments indicate that there is promise in integrating deep learning with argumentative reasoning, resulting in improvements in performance, varying from 1 to 3 percentage points, for determining the truthfulness of a review using a subset of the data set.

We plan to test our deep learning model on the Fake News Challenge (FNC-1), more specifically, determining whether the body text from a news article agrees, disagrees, discusses, or is unrelated to the headline rather than determining whether a news headline supports a tweet as we have done in this article.

Further experimentation is needed to investigate whether the use of argumentative features extracted from BAFs obtained using a deep learning architecture can bring further performance improvements for detecting deceptive reviews. We would like to explore other notions of strength and computed, rather than given, base scores for arguments, when efficient implementations become available to determine whether they affect performance. We would also like to test whether deep learning and argumentative reasoning perform better than standard supervised machine learning techniques in other settings, besides detecting deceptive reviews. Further, we plan to experiment with other deep learning architectures for RbAM. In particular, inspired by the demonstrated effectiveness of attention-based models (Yang et al. 2016; Vaswani et al. 2017), we plan to combine our LSTM-based model with attention mechanisms.

References

Aurisicchio, Marco, Pietro Baroni, Dario Pellegrini, and Francesca Toni. 2015. 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, Buenos Aires.

Baroni, Pietro, Giulia Comini, Antonio Rago, and Francesca Toni. 2017. Abstract games of argumentation strategy and game-theoretical argument strength. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference*, pages 403–419, Nice.

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Transactions on Neural Networks*, 5(2):157–166.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bosc, Tom, Elena Cabrio, and Serena Villata. 2016. Tweeties squabbling: Positive and negative results in applying argument mining on social media. In *Computational Models of Argument, COMMA*, pages 21–32, Potsdam.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642, Lisbon.

Bowman, Samuel R., Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1466–1477, Berlin.

Breiman, Leo. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Cabrio, Elena and Serena Villata. 2012. Natural language arguments: A combined approach. In *ECAI*, volume 242, pages 205–210, Montpellier.

Cabrio, Elena and Serena Villata. 2013. Detecting bipolar semantic relations among natural language arguments with textual entailment: A study. In *Joint Symposium on Semantic Processing (JSSP)* pages 24–32, Trento.

Carstens, Lucas and Francesca Toni. 2015. Towards relation based argumentation mining. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 29–34, Denver, CO.

Carstens, Lucas and Francesca Toni. 2017. Using argumentation to improve classification in natural language problems. *ACM Transactions on Internet Technology*, 17(3):30:1–30:23.

Cayrol, Claudette and Marie-Christine Lagasquie-Schiex. 2005. 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, Barcelona.

Cocarascu, Oana and Francesca Toni. 2016. Detecting deceptive reviews using argumentation. In *Proceedings of the 1st International Workshop on AI for Privacy and Security, PrAISe@ECAI*, pages 9:1–9:8, The Hague.

Cocarascu, Oana and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 1385–1390.

Crawford, Michael, Taghi M. Khoshgoftaar, Joseph D. Prusa, Aaron N. Richter, and Hamzah Al Najada. 2015. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2(1):1–24.

De Smedt, Tom and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research*, 13:2031–2035.

Dean, Jeffrey, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large scale distributed deep networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1223–1231, Lake Tahoe, NV.

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

Dusmanu, Mihai, Elena Cabrio, and Serena Villata. 2017. Argument mining on Twitter: Arguments, facts and sources. In *2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2317–2322, Copenhagen.

Eger, Steffen, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver.

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Fuller, Christie M., David P. Biros, Douglas P. Twitchell, Judee K. Burgoon, and Mark Adkins. 2006. An analysis of text-based deception detection tools. In *12th Americas Conference on Information Systems*, page 418, Acapulco.

Fusilier, Donato Hernández, Manuel Montes-y-Gómez, Paolo Rosso, and Rafael Guzmán-Cabrera. 2015. Detecting positive and negative deceptive opinions using PU-learning. *Information Processing & Management*, 51(4):433–443.

Guo, Weiwei, Hao Li, Heng Ji, and Mona T. Diab. 2013. Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL*, pages 239–249, Sofia.

Habernal, Ivan and Iryna Gurevych. 2016. What makes a convincing argument? Empirical analysis and detecting attributes of convincingness in Web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1214–1223, Austin, TX.

Hai, Zhen, Peilin Zhao, Peng Cheng, Peng Yang, Xiao-Li Li, and Guangxia Li. 2016. Deceptive review spam detection via exploiting task relatedness and unlabeled data. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1817–1826, Austin, TX.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hua, Xinyu and Lu Wang. 2017. Understanding and detecting diverse supporting arguments on controversial issues. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 203–208, Vancouver.

Jindal, Nitin and Bing Liu. 2007. Analyzing and detecting review spam. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 547–552, Omaha, NE.

Kingma, Diederik P. and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Koreeda, Yuta, Toshihiko Yanase, Kohsuke Yanai, Misa Sato, and Yoshiki Niwa. 2016. Neural attention model for classification of sentences that support promoting/ suppressing relationship. In *Proceedings of the Third Workshop on Argument Mining*, pages 76–81, Berlin.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, Lake Tahoe, NV.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Lee, Daniel D. and H. Sebastian Seung. 1999. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791.

Li, Fangtao, Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Learning to identify review spam. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2488–2493, Barcelona.

Li, Jiwei, Myle Ott, Claire Cardie, and Eduard H. Hovy. 2014. 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, Baltimore, MD.

Lim, Ee-Peng, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 939–948, Toronto.

Lippi, Marco and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology*, 16(2):10.

Liu, Pengfei, Xipeng Qiu, Yaqian Zhou, Jifan Chen, and Xuanjing Huang. 2016. Modelling interaction of sentence pair with coupled-LSTMS. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1703–1712, Austin, TX.

Menini, Stefano, Elena Cabrio, Sara Tonelli, and Serena Villata. 2018. Never retreat, never retract: Argumentation analysis for political speeches. In *AAAI*, pages 4889–4896, New Orleans, LA.

Mikolov, Tomas, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural

network based language model. In *INTERSPEECH*, pages 1045–1048, Makuhari.

Mukherjee, Arjun, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malú Castellanos, and Riddhiman Ghosh. 2013. Spotting opinion spammers using behavioral footprints. In *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, pages 632–640, Chicago, IL.

Mukherjee, Arjun, Bing Liu, and Natalie Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web*, pages 191–200, Lyon.

Mukherjee, Subhabrata, Sourav Dutta, and Gerhard Weikum. 2017. Credible review detection with limited information using consistency analysis. *CoRR*, abs/1705.02668.

Niculae, Vlad, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver.

Ott, Myle, 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*, pages 497–501, Atlanta, GA.

Ott, Myle, 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*, pages 309–319, Portland, OR.

Pedregosa, Fabian, 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 Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha.

Poddar, Lahari, Wynne Hsu, and Mong-Li Lee. 2017. Author-aware aspect topic sentiment model to retrieve supporting opinions from reviews. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 483–492, Copenhagen.

Potash, Peter, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1364–1373, Copenhagen.

Rago, Antonio, Francesca Toni, Marco Aurisicchio, and Pietro Baroni. 2016. 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, Cape Town.

Rago, Antonio, Francesca Toni, and Pietro Baroni. 2018. How many properties do we need for gradual argumentation? In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1736–1743, New Orleans, LA.

Ren, Yafeng and Donghong Ji. 2017. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385:213–224.

Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Rout, Jitendra Kumar, Smriti Singh, Sanjay Kumar Jena, and Sambit Bakshi. 2017. Deceptive review detection using labeled and unlabeled data. *Multimedia Tools and Applications*, 76(3):3187–3211.

Sandulescu, Vlad and Martin Ester. 2015. Detecting singleton review spammers using semantic similarity. In *Proceedings of the 24th International Conference on World Wide Web*, pages 971–976, Florence.

Schuster, Mike and K. Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *Transactions on Signal Processing*, 45(11):2673–2681.

Shojaee, Somayeh, Masrah Azrifah Azmi Murad, Azreen bin Azman, Nurfadhlina Mohd Sharef, and Samaneh Nadali. 2013. Detecting deceptive reviews using lexical and syntactic features. In *13th International Conference on Intelligent Systems Design and Applications, ISDA*, pages 53–58, Salangor.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Stevens, Keith, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961, Jeju Island.

Talathi, Sachin S. and Aniket Vartak. 2015. Improving performance of recurrent neural network with ReLU nonlinearity. *CoRR*, abs/1511.03771.

Tan, Stephanie. 2017. Spot the lie: Detecting untruthful online opinion on Twitter. `https://www.doc.ic.ac.uk/~oc511/reportStephanie.pdf`.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 6000–6010, Long Beach, CA.

Viviani, Marco and Gabriella Pasi. 2017. Quantifier guided aggregation for the veracity assessment of online reviews. *International Journal of Intelligent Systems*, 32(5):481–501.

Xie, Sihong, Guan Wang, Shuyang Lin, and Philip S. Yu. 2012. 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, Beijing.

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, CA.

Yin, Wenpeng, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *TACL*, 4:259–272.

Zhao, Weizhong, James J. Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. 2015. A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC Bioinformatics*, 16(13):S8.

Zhou, Lina, Judee K. Burgoon, Jay F. Nunamaker, and Doug Twitchell. 2004. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group Decision and Negotiation*, 13(1):81–106.