Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

Combining deep learning and argumentative reasoning for the analysis of social media textual content using small datasets*

Oana Cocarascu
Imperial College London

Francesca Toni
Imperial College London

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 analysing whether news headlines support tweets and whether reviews are deceptive by analysing 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 towards a statement is a useful step towards 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 datasets.*

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 analysing whether news headlines support tweets and whether reviews are deceptive by analysing 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 datasets (Dean et al. 2012; Krizhevsky, Sutskever, and Hinton 2012; Lecun,

*

Submission received: October 13, 2017
Revised version received: May 25, 2018
Accepted for publication: August 19, 2018

Bengio, and Hinton 2015; Silver et al. 2016). In this 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 datasets. 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 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 analysed. 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 which introduced the corpus used in this article (Carstens and Toni 2015, 2017)[1].

We then test our best performant deep learning model on different datasets consisting of news articles headlines to determine whether these *support* tweets, and show that our model generalises well. We use two datasets introduced in Tan (2017): one consisting of pairs of tweets–headlines related to FBI's investigative involvement in Hillary Clinton's email leak and the second one adapted from Guo et al. (2013). For example, for

[1] See https://www.doc.ic.ac.uk/~lc1310/.

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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 towards a statement is a useful step towards determining its truthfulness.

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."*

Then, 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.

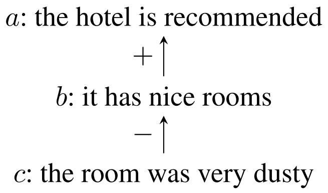


Figure 1
Example BAF. Here, b supports a (indicated as a + edge) and c attacks b (indicated as a - edge).

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

[2] http://www.fakenewschallenge.org/

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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 to 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 dataset from Ott, Cardie, and Hancock (2013); Li et al. (2014). These are the gold standard in deception detection for reviews but are rather small (1600 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 dataset when using a subset of the dataset.

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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 modelling 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 dataset issue. Albeit small, the improvements show promise in the integration of deep learning and symbolic, argumentative reasoning.

The remainder of this article is organised 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 datasets 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 datasets 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 (AM)

Existing 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 (CNNs) to model pairs of sentences in tasks such as textual entailment and answer selection. Determining the relations in a SNLI (Stanford Natural Language Inference) 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. Recognising textual entailment between two sentences is also addressed by Rocktäschel et al. (2015), using LSTMs and word–by–word neural attention mechanism on the SNLI dataset. Liu et al. (2016) propose two models that capture the interdependencies between two parallel LSTMs encoding the two sentences for the tasks of recognising 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 dataset of 1000 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 initialised 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 1462 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, support as in our work. Notably, Habernal and Gurevych (2016) experiment with both Bidirectional LSTMs (BiLSTM) and BiLSTM extended with an attention mechanism and a convolution layer over the input to determine the class which explains why a certain argument is more convincing than the other in the pair. Whilst 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, 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 (Xie et al. 2012).

Different forms of machine learning have been used in the literature to detect deceptive behaviour, notably unsupervised (Mukherjee et al. 2013), semi–supervised (Fusilier et al. 2015) and supervised (Ott et al. 2011; Li et al. 2014; Ott, Cardie, and Hancock 2013; Shojaee et al. 2013) 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 (Li et al. 2011; Jindal and Liu 2007; 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 (Fuller et al. 2006; Zhou et al. 2004).

Hai et al. (2016) use review spam detection for different domains (hotel and restaurant) as a multi–task 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 datasets used in this article.

3.1 Recurrent Neural Networks (RNNs) and variations

RNNs (Elman 1990; Mikolov et al. 2010) are a type of neural networks 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 RNNs which use memory cells to store contextual information and three types of gates (input, forget, and output gates) that determine which 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

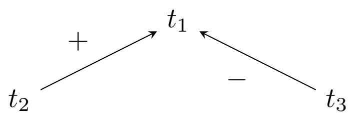
(Abstract) Argumentation Frameworks (AAFs), introduced by Dung (1995), are pairs consisting of a set of arguments and a binary relation between arguments, representing attacks. Formally, an AAF is any $\langle AR, attacks\rangle$ where $attacks \subseteq AR \times AR$. BAFs extend AAFs by considering two independent binary relations between arguments: attack and support (Cayrol and Lagasquie-Schiex 2005). Formally, a BAF is any $\langle AR, attacks, supports\rangle$ where $attacks \subseteq AR \times AR$ and $supports \subseteq AR \times AR$. 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. Like 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 Discontinuity–Free Quantitative Argumentation Debate (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 which 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 Dataset

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

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 recognising textual entailment, contradiction, and neutral relations) that we ultimately decided not to include due to their structure not being amenable to our analysis, for the reasons below.

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. Whilst we initially hypothesised 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 labelled 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

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

of sentence pairs found in this corpus is different from the types of texts we are interested in analysing 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. For example 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 Datasets

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).

The hotel dataset that we use 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 (Ott, Cardie, and Hancock 2013). The restaurant dataset 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 stand–alone 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) CNNs (Yin et al. 2016; Habernal and Gurevych 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. Whilst we do not impose texts to be single sentences, 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 initialise 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 respectively. Indeed, based on our assumption that if one sentence attacks/supports another sentence, then both may be considered to be argumentative, irrespective of their stand–alone argumentativeness, we opted for two classifiers to model the two texts independently of one another, and then merge them.

[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.

Each (Bi)LSTM model produces a vector representation of the text being analysed 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 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 element–wise 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.

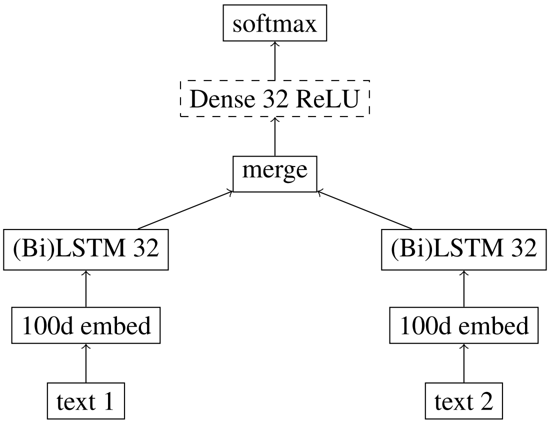


Figure 2
Our classification architecture: two LSTMs are run with one text each. We tested two types of LSTMs: unidirectional and bidirectional. The dashed layer (Dense 32 ReLU) is optional and its inclusion in the architecture and impact on performance was tested in our experiments.

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, amongst 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 optimised using the Adam method (Kingma and Ba 2014) with learning rate 0.001. Indeed this method gave better performances than alternative optimisers 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.

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

Table 1
Hyper–parameters for our LSTM and BiLSTM models.

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

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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 2
5x10 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 5x10 fold cross–validation.

We run 10 stratified fold cross–validation (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 (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 give 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

Baseline	**A%**	**P%**	**R%**	**$\mathbf{F_1}$%**						
LR (unigrams)	77.87	78.02	77.87	77.89						
Model/	**Non–trained embeddings**				**Trained embeddings**					
Merge/Dense	**A%**	**P%**	**R%**	**$\mathbf{F_1}$%**	**A%**	**P%**	**R%**	**$\mathbf{F_1}$%**	**Astd**	**$\mathbf{F_1}$std**
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

Table 3
5x10 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 5x10 fold cross–validation.

dataset and that this effect could be diminished by acquiring 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 to 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 to the baseline. Training the word embeddings results in better performance compared to 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 to 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 whether our proposed model correctly identifies that a headline of a news article *supports* a tweet. We use the two datasets introduced in Tan (2017)[6]: one consisting of pairs of tweets–headlines related to FBI's investigative involvement in Hillary Clinton's email 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 NY Times. The authors extracted the news titles from the urls with the aim to find the most relevant article to the tweet.

We used the filtered dataset 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 dataset can be seen in Table 4.

[6] https://www.doc.ic.ac.uk/~oc511/data.json

Dataset	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 4
Examples from the tweet datasets.

Dataset	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

Table 5
Performance of our model on the tweet datasets.

The performance of our model described in Section 4 on the two tweet datasets is given in Table 5. On the tweet datasets consisting of 30 and 840 examples our model yields 73% F_1 and 94% F_1 respectively, thus it generalises 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 towards a statement is a useful step towards determining its truthfulness.

[7] http://www.fakenewschallenge.org/

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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 to a topic, to more advanced techniques related to topic modelling 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 which 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 below:

1. Split each review into sentences (where each sentence is a potential argument)

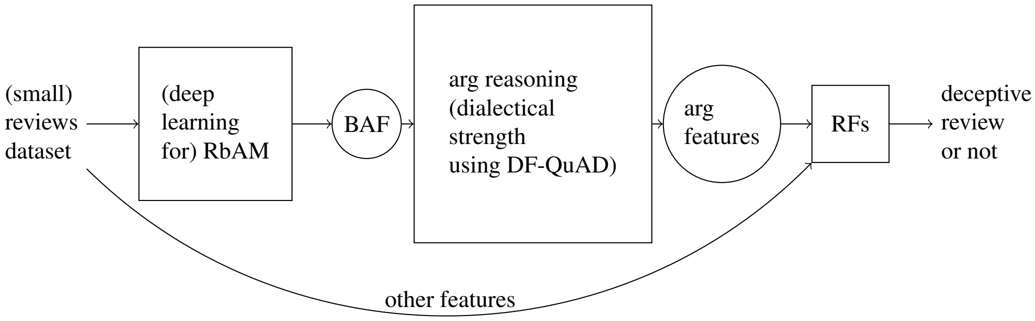


Figure 3
Overview of how (deep learning for) RbAM and argumentative (arg) reasoning and features are used for detecting deceptive reviews.

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 to this topic to determine the relations between them
4. Construct BAF

This process is done for the set of reviews associated to each *item* in the review (i.e. the hotel dataset 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 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). 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. We use LDA with online variational Bayes algorithm.

[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"*.

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

Each document is represented as a bag of words where the vocabulary represents the set of words in the dataset. 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 and we use NMF with 0.1 regularization to optimise 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 analysing 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 to 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 to 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 amongst 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.

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)

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

For our deep learning architecture, in order to identify the relations between arguments associated to 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 G. In addition to these special arguments G_t supporting G, the BAFs also include relations between arguments related to topic t drawn from 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:

$$AR = \{G, G_{room}, G_{food}, G_{service}, G_{Wi-Fi}, a_{11}, a_{12}, a_{21}, a_{22}, a_{23}\},$$

$$attacks = \{(a_{12}, G_{food}), (a_{23}, G_{Wi-Fi})\}$$

$$supports = \{(a_{22}, a_{11}), (a_{11}, G_{room}), (a_{21}, G_{service}),$$

$$(G_{room}, G), (G_{food}, G), (G_{service}, G), (G_{Wi-Fi}, G)\}$$

shown graphically in Figure 4 (where edges labelled - represent attacks and edges labelled + 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

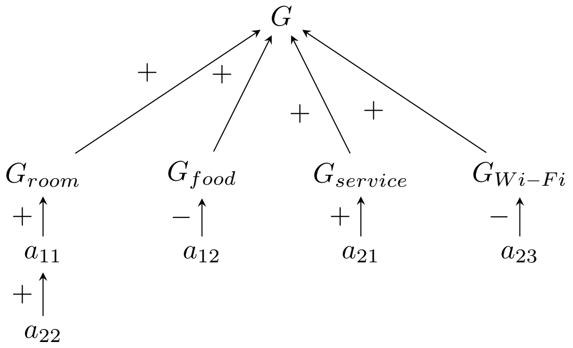


Figure 4
Topic–dependent BAF obtained from r_1,r_2 in Section 6.1.1.

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 to limit the 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.

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 to 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$\setminus\{r\}$.

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.

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

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

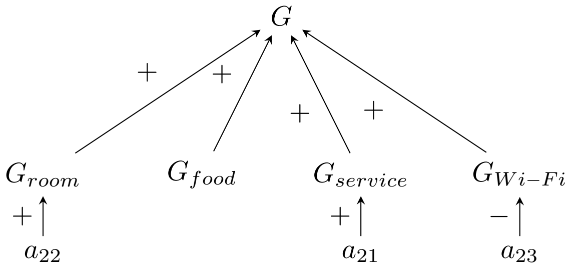


Figure 5
BAF obtained from removing (arguments from) r_1.

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 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 summarised in Table 7.

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

Table 7
Features and the associated category.

Additionally, we include *tf–idf* (term frequency–inverse document frequency) features obtained from all reviews using *scikit–learn* (Pedregosa et al. 2011). To calculate these, we use the lemmas obtained by analysing 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 modelling, 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).

For each approach of topic modelling, we identify the topic that has the highest probability of being associated to the review, as well as all the topics with probability greater than 0.2 of of being associated to the review. In both cases, we extract the sentences that contain any of the top words related to the topics that have been associated to the review. To identify the relations between arguments associated to each topic, we chose the best performing instance of our deep neural architecture trained on the full RbAM dataset. 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 dataset are shown in Table 8. We see that adding the *tf–idf* features gives 76% accuracy, resulting in a dramatic improvement of 12% compared to the baseline where the syntactic features from Table 7 were used. Using argumentative features extracted from the BAF constructed from topics being associated to 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 modelling 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 modelling rather than simple associations of topics $\sim$ nouns, and LSTMs for RbAM, with 0.38% improvement compared to using syntactic features and *tf–idf* features.

Standard features	**Unigrams**	**Argumentative features**	**Topic model**	**RFs**	
				Accuracy	$\mathbf{F_1}$
✓	✗	✗	✗	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 8
Classifiers' performances on the hotel dataset.

The classifiers' performances on the restaurant dataset are shown in Table 9. Here as well, we see that adding the *tf–idf* features results in a dramatic improvement of 9% compared to the baseline. In contrast to the hotel dataset, using argumentative features extracted from the BAF constructed from topics being associated to nouns in reviews and using RFs for RbAM results in an improvement of 1.5% compared to using only syntactic features. Using argumentative features extracted from the BAF constructed using LDA and NMF for topic modelling 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 modelling and LSTMs for RbAM, with 2.75% improvement compared to using syntactic features and *tf–idf* features.

Standard features	**Unigrams**	**Argumentative features**	**Topic model**	**RFs**	
				Accuracy	$\mathbf{F_1}$
✓	✗	✗	✗	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

Table 9
Classifiers' performances on the restaurant dataset.

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.

We did not carry out any direct comparison with the results documented in the papers that introduced the reviews datasets 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 dataset used in this article whereas Ott, Cardie, and Hancock (2013) focus on classifier performances on the hotel reviews dataset 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 multi–class classification tasks, with the aim to classify reviews based on their source (reviews written by customers, employees, turkers).

7. Deep learning and Argumentation on small datasets

The datasets described in Section 3.4 and used in this article differ in size: the hotel dataset contains 1600 reviews whereas the restaurant dataset contains 400 reviews. We obtained improvements compared to the baseline as presented in Section 6.3 when using more advanced topic modelling techniques and LSTMs to determine relations between arguments extracted from the reviews.

The datasets used are small. Moreover, the size of the restaurant dataset represents a quarter of the size of hotel dataset. We are interested to see whether, using a subset of the hotel dataset, we can still achieve comparable or better results compared to using the entire dataset 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 datasets.

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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 instance of our deep neural architecture trained on the full RbAM dataset. We see that, using a subset of the hotel dataset, we achieve better results combining deep learning with argumentative reasoning compared to using the entire dataset. We obtain 1% improvement in accuracy when using a quarter of the dataset, and 3% accuracy improvement when using three quarters or half of the hotel dataset. 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).

Total number of reviews	**RFs**	
	Accuracy	**F_1**
1600	76.69	76.48
1200	79.83	79.74
800	79.63	79.48
400	77.75	77.62

Table 10
Classifier performance on the hotel dataset using subsets of the dataset.

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 dataset 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 datasets consisting of news articles headlines that *support* tweets and showed that our model generalises well. This suggests our model can be 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, Springer.

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.

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.

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)*, Association for Computational Linguistics.

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*.

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, IOS Press.

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)*.

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

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, Springer Berlin Heidelberg.

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, ACM.

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, TaghiM. Khoshgoftaar, JosephD. Prusa, AaronN. 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, Curran Associates Inc.

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 *The 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, Copenhagen, Denmark.

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, Association for Computational Linguistics.

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, Association for Information Systems.

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.

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.

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.

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.

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, IEEE Computer Society.

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*.

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, Curran Associates Inc.

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, AAAI Press.

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, Association for Computer Linguistics.

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, ACM.

Lippi, Marco and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Techn.*, 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.

Menini, Stefano, Elena Cabrio, Sara Tonelli, and Serena Villata. 2018. Never retreat, never retract: Argumentation analysis for political speeches. In *AAAI*.

Mikolov, Tomas, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

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

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, ACM.

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, Association for Computational Linguistics.

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*, Association for Computational Linguistics.

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, Association for Computational Linguistics.

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.

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.

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.

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, AAAI Press.

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*.

Ren, Yafeng and Donghong Ji. 2017. Neural networks for deceptive opinion spam detection: An empirical study. *Inf. Sci.*, 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 Appl.*, 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, ACM.

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 Intellient Systems Design and Applications, ISDA*, pages 53–58, IEEE.

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

Computational Linguistics Just Accepted MS.
doi:10.1162/coli_a_00338

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, Association for Computational Linguistics.

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.

Viviani, Marco and Gabriella Pasi. 2017. Quantifier guided aggregation for the veracity assessment of online reviews. *Int. J. Intell. Syst.*, 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, ACM.

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.

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. In *BMC Bioinformatics*.

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.