Attending Sentences to detect Satirical Fake News

Sohan De Sarkar
Dept. of Computer Science
Indian Institute of Technology
Kharagpur, West Bengal, India
sohandesarkar@gmail.com

Fan Yang
Dept. of Computer Science
University of Houston
3551 Cullen Blvd., Houston
fyang11@uh.edu

Arjun Mukherjee
Dept. of Computer Science
University of Houston
3551 Cullen Blvd., Houston
arjun4787@gmail.com

Abstract

Satirical news detection is important in order to prevent the spread of misinformation over the Internet. Existing approaches to capture news satire use machine learning models such as SVM and hierarchical neural networks along with hand-engineered features, but do not explore sentence and document difference. This paper proposes a robust, hierarchical deep neural network approach for satire detection, which is capable of capturing satire both at the sentence level and at the document level. The architecture incorporates pluggable generic neural networks like CNN, GRU, and LSTM. Experimental results on real world news satire dataset show substantial performance gains demonstrating the effectiveness of our proposed approach. An inspection of the learned models reveals the existence of key sentences that control the presence of satire in news.

1 Introduction

In the era of the Internet, online journalism is now a common practice. Online news articles have a major contribution in keeping people informed about what is happening in the world. The usage of Internet to spread news comes with the disadvantage of deception. The presence of deceptive and misleading news articles has been around for a while. Although some news articles often have a disclaimer about it being fake, many other don't and thus readers could be led to believe them to be true. This leads to spread of misinformation, which may also start off a rumour. The importance of the detection of deceptive news is increasing rapidly, as more and more people start relying on online news as their major source of news.

News satire is a genre of deceptive news that is found on the web, with the intent of dispensing satire in the form of legitimate news articles. These articles differ from "fake" news, in the sense that fake news intend to mislead people by providing untrue facts, while satirical news intends to ridicule and criticize something by providing satirical comments or through fictionalized stories. Satire is the intention of the author to be discovered as "fake", unlike fake news, in which the intention is to make make the readers believe in the news as true. Detection of news satire is thus important to control the spread of false stories.

We propose a hierarchical deep neural network model for satirical news detection, that is able to capture satire both at the sentence level and at the document level. The architecture is very extensible and caters to a variety of plug-and-play neural network models such as CNN, LSTM and GRU. This pipelined architecture allows for optimal learning of parameters required to capture satire. We show that our model is able to capture satire more efficiently than existing models, by using only pretrained word embeddings as input, without the aid of any syntactic information or any hand-crafted features. We show that word level semantic information is sufficient for effective detection of satire, with word level syntax information only marginally improving the performance. An analysis of the learned models reveals that news satire is decided by a few key sentences of the news article, the last sentence being one of them.

We use the dataset introduced in (Yang et al., 2017) as the dataset for satire news detection. We trained our proposed plug-and-play hierarchical model end-to-end on the ground truth data. Our model works at the sentence level as opposed to paragraph level attention in (Yang et al., 2017). We perform extensive

Proceedings of the 27th International Conference on Computational Linguistics, pages 3371–3380
Santa Fe, New Mexico, USA, August 20-26, 2018.

experiments on the dataset, fine-tuning the model by plugging different neural network models into the architecture. Experimental results on the dataset shows superior performance of our model compared to existing state-of-the-art approaches.

2 Related Work

Previous approaches for generic deception detection include the use of traditional machine learning model such as SVM (Zhang et al., 2012) and Naive Bayesian models (Oraby et al., 2015). These approaches focus on using linguistic cues and the social network behavior (Conroy et al., 2015) to detect deception. Much work has been done for deception detection on social media platforms (Davidov et al., 2010; Reyes et al., 2012) and opinion spam (Ott et al., 2011; Mukherjee et al., 2012; Mukherjee et al., 2013) In the context of deceptions in news, the field of "fake" news detection has been explored before (Jin et al., 2016; Rubin et al., 2016). These also include the use of machine learning, some of them also leveraging neural networks (Wang, 2017; Ruchansky et al., 2017) for the task.

Existing works towards satirical news detection focus on engineering features to denote satire. (Burfoot and Baldwin, 2009) filter satirical news from true news with headline features, profanity, and slang. (Rubin et al., 2016) propose additional features to classify satirical news, including absurdity, humour, grammar, negative affect, and punctuation. (Yang et al., 2017) further show linguistic features could be incorporated at paragraph level and reveal the different behaviour of each feature at paragraph level and document level. These models heavily rely on linguistic/word features as opposed to our representation learning approach. From these works, we observe that word level features contribute to the detection most while linguistic features only improve the result by a little, so we focus on our model to detect satire without further hand-crafted features.

While features generated with careful hand analysis might contribute a robust classifier, neural network based models, from convolutional neural network (Kim, 2014; Kalchbrenner et al., 2014) to recurrent neural network (Tang et al., 2015), or a hybrid of the two (Lai et al., 2015), have pushed classification task to a new level. Also, the recent advances in learning distributed representations for word semantics in the form of word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2016) allow for better modeling of semantics both at the sentence and document level. In this work, we utilize the power of neural networks and aim to advance the result of satirical news detection. We pack two separate composition models to further enhance the performance of the learned representation.

3 Model

We propose an approach for building a robust hierarchical neural network architecture for detecting satire news, as shown in Figure 1. We abstract the whole network into two major components, the S and D module. The compositional module S creates a sentence embedding, taking a sequence of word embeddings as inputs. The compositional module D creates a document embedding, which acts as a summarization of the document, taking sentence embeddings as input. We use the learned document embeddings to classify the news as satire or true. This kind of abstraction helped us to fine-tune the architecture by applying different choices of compositional models for the S and D module.

3.1 Word embeddings and Syntax

We use different pretrained word embeddings such as Glove[1] (Pennington et al., 2014) and fastText[2] (Bojanowski et al., 2016) as the initial word embeddings. These pretrained embeddings are (optionally) concatenated with one-hot embeddings that contain the syntax information[3] of the word (Baccianella et al., 2010; Miller, 1995). The various syntactic features used and their corresponding one-hot vector lengths is shown in Table 1. The named entities used are: FACILITY, GPE, GSP, LOCATION, ORGANIZATION, PERSON, NULL(representing no named entity). The SentiWordnet scores are 16 discrete values ranging between 0 and 1, thus requiring a one-hot vector of size 16 to represent each score. These word

[1] https://nlp.stanford.edu/projects/glove

[2] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[3] http://www.nltk.org

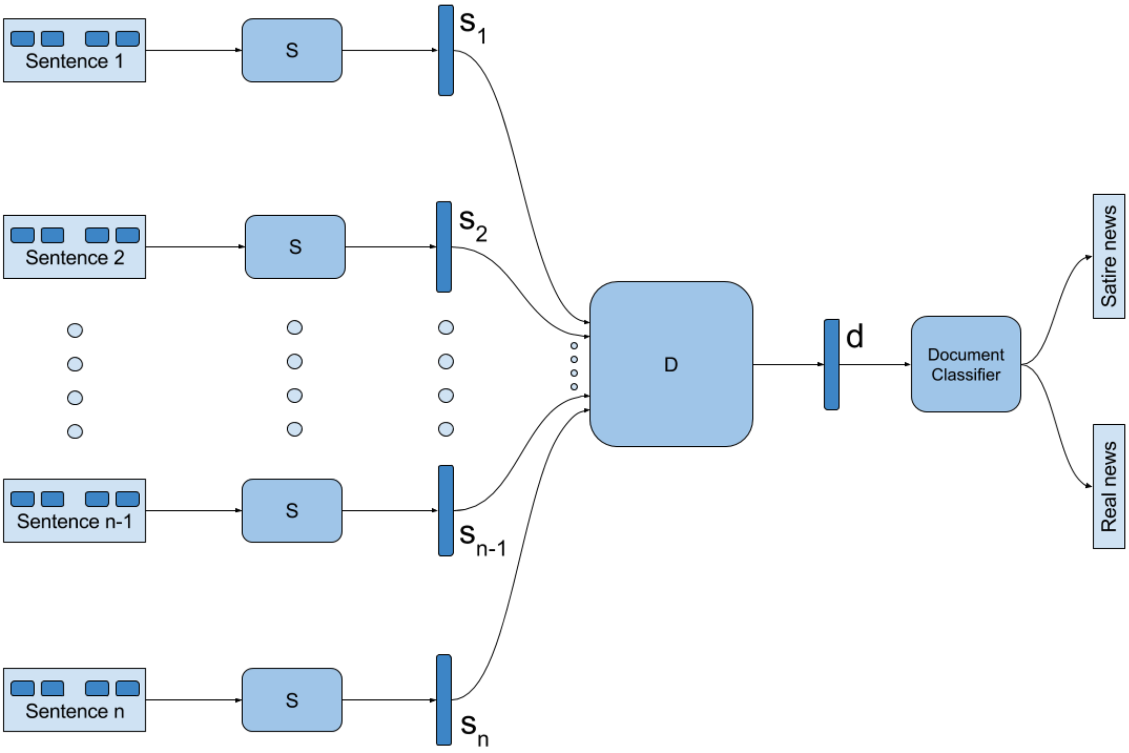


Figure 1: Model Architecture

embeddings (concatenated with syntax information) are multiplied with a weight matrix W_{emb} (learned) to produce a final word embedding, that summarizes the required semantics of the word for capturing satire.

3.2 Sentence (S) Module

The S module takes a sequence of word embeddings as input, and produces a sentence embedding. This module tries to capture the essential information for capturing satire in a news at the sentence level. The various model choices for the S module include Temporal Convolutional neural networks(Kim, 2014) (CNN) and sequential models like Long Short-Term Memory(Hochreiter and Schmidhuber, 1997) (LSTM) and Gated Recurrent Unit(Cho et al., 2014) (GRU).

Let $v_i \in \mathbb{R}^d$ be the d-dimensional word embedding of the i^{th} word of a sentence of length n. We show 3 different models to produce a sentence embedding from the word embeddings. Then, the S module can be represented mathematically as a composition function f that takes a sequences of n word embeddings as input to produce a sentence embedding s. Thus,

$$s = f([v_1, v_2, \dots v_n]) \tag{1}$$

where the choice of the composition function are standard generic neural networks like $LSTM, GRU, CNN$. In the case of LSTM/GRU, we use their deep bidirectional versions, where we stack multiple bidirectional LSTM/GRU on top of each other.

3.3 Document (D) Module

Similar to the S module, D module takes a sequence of sentence embeddings as input and produces a document embedding, capturing information at the document level. The embedding produced by this

Syntax feature	Length
Part-Of-Speech tag	44
SentiWordnet positive score	16
SentiWordnet negative score	16
SentiWordnet objective score	16
Named Entity IOB tag	3
Named Entity tag	7
Topmost Wordnet synset same as word synset	2
Starts with uppercase letter	2
All uppercase letters	2
Is number	2

Table 1: Word level syntax features and their one-hot vector lengths

module is directly used for classification. The choice of compositional models for the D module are the same as those for the S module, i.e. Temporal CNN and sequential neural networks like GRU and LSTM, with the difference that it takes sentence embeddings as input instead of word embeddings.

3.3.1 Attention Layer

This layer performs a weighted average of its inputs, where the weights are learned by a neural network, and serve as attention weights (Bahdanau et al., 2014) for the sentences. Let the outputs of the D module be $\{o_1, o_2, \ldots o_n\}$ for a document containing n sentences. We use a two hidden layer neural network for obtaining the attention weights. We obtain the attention weights, $\hat{a}$, as follows

$$a_i = W_a^{(2)} \cdot \tanh(W_a^{(1)} \cdot o_i) \tag{2}$$

$$a = [a_1, a_2, \ldots a_n] \tag{3}$$

$$\hat{a} = softmax(a) \tag{4}$$

Here, $W_a^{(1)} \in \mathbb{R}^{mk}$ and $W_a^{(2)} \in \mathbb{R}^k$ are weight matrices (trained), with m being the size of the output embeddings and k being the size of the hidden layer of the neural network. The final document embedding d is taken as a weighted sum of the outputs, i.e. $d = \sum_{i=1}^{n} \hat{a}_i \cdot o_i$.

3.3.2 Attentive Concatenate Layer

This layer is applied before the application of the compositional model of the D module, taking input the sentence embeddings obtained from the S module. The layers applies attention over the sentences in context of a particular sentence. Let the sentence embeddings of the sentences be $\{s_1, s_2, \ldots s_n\}$ in a document of n sentences. Let A be the attention module presented above. This layer applies A over the sentence embeddings after concatenating the context sentence embedding, to obtain a weighted average embedding, which is concatenated with context sentence embedding to obtain the output, $o_i = [s_i, t_i]$.

$$t_i = A(\{[s_i, s_1], [s_i, s_2], \ldots [s_i, s_n]\}) \tag{5}$$

3.4 Document Classifier

The final document classifier is a neural network with two hidden layers followed by a softmax layer. This takes a m dimensional document vector, d as a input, and produces a label, l, as Real(0) or Satire(1).

$$l = \arg\max(softmax(W_d^{(2)} \cdot f(W_d^{(1)} \cdot d))) \tag{6}$$

Here, f is the ReLU activation function. $W_d^(1) \in \mathbb{R}^{mk}$ and $W_d^{(2)} \in \mathbb{R}^{2k}$ are the weight matrices.

4 Experimental Evaluation

We now detail the evaluation. We first detail the experiment settings, followed by baselines and finally the results.

4.1 Dataset and Preprocessing

	#Train	#Vali	#Test	#Sent	#Word	#Digit
True	101,268	33,756	33,756	32	734	93
Satire	9538	3,608	3,103	25	587	49

Table 2: The split of the dataset and the average count of sentences, words, and digits per document.

4.1.1 Dataset

We utilize the dataset from (Yang et al., 2017). The satirical news were originally collected from the 14 satirical websites, including Onion, theSpoof, SatireWorld, Beaverton, Ossurworld, DailyCurrent, DailyReport, EnduringVision, Gomerblog, NationalReport, SatireTribune, SatireWire, Syruptrap, and UnconfirmedSources. While the true news is collected from Google News and major news outlets, including CNN, Dailymail, WahingtonPost, NYTimes, TheGuardian, and Fox. For true news, we did not split sources. For satirical news, we used two most popular satirical websites, Onion and theSpoof, which have the largest number of satirical news, for training, while the rest of the sources were chosen randomly for test and validation to yield a richer evaluation set. The split and the description of the dataset can be found in Table 2.

4.1.2 Preprocessing

Sentences occurring twice or more are removed from the dataset. Also, each sentence in the dataset is padded to a length of 100, with a special <PAD> word. For sentences having more than 100 words, we consider the sentence as only the first 100 words. Similarly, each news article (or document) is padded to a length of 100 sentences by adding extra sentences containing the <PAD> word. For documents having more than 100 sentences, we consider only the first 100 sentences.

4.2 Experiment Settings

The whole model is trained end-to-end, i.e. all the parameters (weights) of the model are conditioned on the response variable (True news or Satire news). We experimented with different models by changing the choice of word embeddings, the S module, and the D module. For each model, we trained it for 20 epochs with Mini-batch Stochastic Gradient Descent algorithm. Training loss function used was cross entropy. For learning the optimal parameters for each model, we optimized on either the validation accuracy or the validation F1-score based on whichever performed better (these are listed in Table 3). Sentence embedding size and document embedding size were fixed at 300, for all the models. Word embedding (summarized) size was 300 for all models.

The word embedding for the <PAD> was set to all-zero embedding. We used the word embedding for “unk” as the word embedding for all words whose word embeddings were not available. We used the word embedding for “#” to represent all numbers.

4.3 Baselines

We compare our model with 4 baselines:

- **SVM word+char ngrams**: 1,2-word grams plus bigrams and trigrams of the characters
- **Rubin et al**(Rubin et al., 2016): Unigram and bigrams tf-idf with satirical features proposed in their works. It reports a better result than (Burfoot and Baldwin, 2009) so we omit the comparison with the latter.
- **Le et al**(Le and Mikolov, 2014): Unsupervised method learning distributed representation for document
- **Yang et al**(Yang et al., 2017): A 4 level hierarchical network considering charactors, words, paragraphs, and documents. The also incorporate 4 families of linguistic features at both paragraph level and document level.

4.4 Results and Discussion

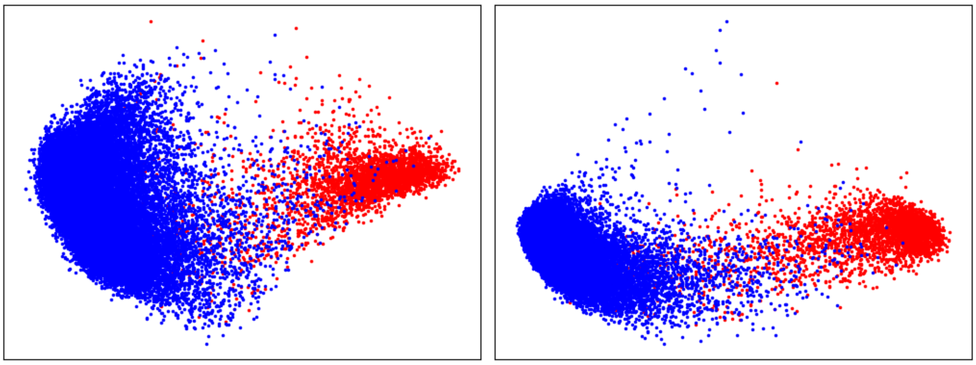
S module	D module	Max	Validation				Test			
			Acc	P	R	F	Acc	P	R	F
Baselines										
SVM word + char ngrams		Val F1	97.43	87.10	81.57	84.24	97.64	90.76	84.12	87.31
Rubin et al (Rubin et al., 2016)		Val F1	97.73	90.21	81.92	85.86	97.79	93.47	82.95	87.90
Le et al (Le and Mikolov, 2014)		Val F1	92.48	58.48	71.66	64.40	90.48	50.52	67.88	57.92
Yang et al (Yang et al., 2017)		Val F1	**98.54**	**93.31**	**89.01**	**91.11**	**98.39**	**93.51**	**89.50**	**91.46**
Word embeddings used: One-hot										
D-B-GRU	A-D-B-GRU	Val F1	97.77	91.78	84.56	88.02	97.96	89.49	85.94	87.68
Word embeddings used: Glove										
CNN	Average	Val acc	97.43	90.86	81.62	85.99	97.77	88.09	85.11	86.57
CNN	CNN	Val acc	97.45	87.45	86.00	86.72	97.76	84.92	**89.30**	87.05
CNN	B-LSTM	Val acc	97.81	95.32	81.37	87.79	98.02	94.22	81.50	87.40
D-B-GRU	A-D-B-GRU	Val acc	98.02	**95.93**	83.09	89.05	98.28	**94.97**	84.04	89.17
D-B-GRU	D-B-GRU	Val acc	98.00	93.05	85.78	89.27	98.23	91.03	87.65	89.31
CNN	A-D-B-GRU	Val acc	97.99	93.67	84.97	89.11	98.26	92.15	86.72	89.35
D-B-LSTM	A-D-B-LSTM	Val F1	**98.18**	94.15	**86.55**	**90.19**	98.31	93.45	86.01	89.57
B-GRU	B-GRU	Val acc	97.90	94.83	82.87	88.44	98.42	94.43	86.43	90.25
CNN	A-B-LSTM	Val F1	97.99	95.42	83.23	88.91	**98.57**	94.20	88.49	**91.25**
Word embeddings used: Glove + fastText										
D-B-LSTM	A-D-B-LSTM	Val F1	97.67	93.49	81.65	87.17	98.30	93.04	86.27	89.53
CNN	A-D-B-GRU	Val F1	**98.06**	92.31	**87.19**	**89.68**	98.39	90.98	**89.81**	90.39
B-GRU	A-D-B-GRU	Val F1	98.01	**95.63**	83.23	89.00	**98.63**	**95.20**	88.20	**91.56**
Word embeddings used: Glove + Syntactic Information										
CNN	A-B-GRU	Val F1	97.81	90.79	**86.08**	**88.37**	97.91	87.40	87.88	87.64
D-B-LSTM	A-D-B-LSTM	Val F1	97.48	89.17	84.22	86.63	98.10	87.77	90.04	88.89
CNN	A-D-B-GRU	Val F1	**97.84**	**93.02**	83.95	88.25	**98.59**	**92.02**	**91.16**	**91.59**

Table 3: Results. Prefix notation; **A: Attentive, D: Deep (3 layers), B: Bidirectional**

The results of the various experiments performed by us, using different choices of word embeddings, S module and D module is summarized in Table 3. Our best model outperforms the baseline models on the dataset. We observe that adding word level syntax information improves the performance only by a small margin. Thus, we can conclude that at the word level, semantic information is more relevant to capture satire than syntax information. For further analysis, we shall refer to the (CNN, A-B-LSTM) model using only Glove embeddings as Model A, and the (B-GRU, A-D-B-GRU) model using Glove and fastText embeddings as Model B.

4.4.1 Analysis of D-module

Figure 2 shows a PCA decomposition of the document embeddings learned by the models, on the test data. We see that satirical news (in red) and real news (in blue) form two separate clusters in both the models. This clearly shows that the embeddings learned by the D module successfully captures satire.



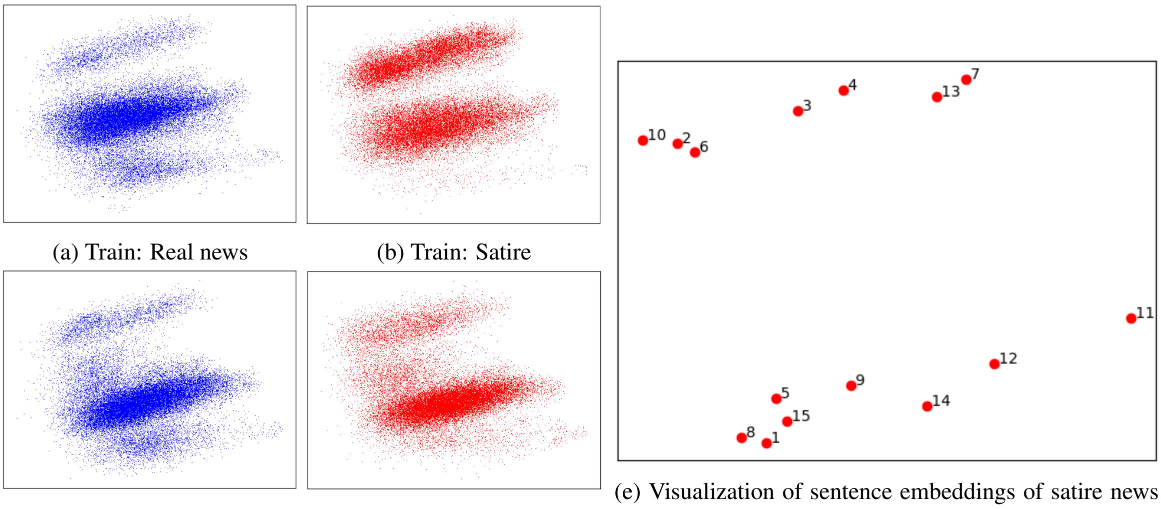
(a) Model A

(b) Model B

Figure 2: Visualization of document embeddings of true(blue) and satire(red) news using PCA

4.4.2 Analysis of S-Module

Applying PCA decomposition on the sentence embeddings learned by Model A reveals a few interesting properties that may be relevant in capturing satire. Figure 3 shows the PCA decomposition of sentence embeddings of news articles from the train and the test data. A closer look at the plots of the train data (Figures 3a) reveal that the sentence embeddings form three clusters. Most of the sentences lie in the central cluster, while some lie the cluster above the central one. A third cluster is observed below the central cluster, and consists mainly of sentences from real news articles. We also note that the density of the upper cluster is higher for satirical news sentences (visibly evident only in the training data), as compared to that for real news. We find that the values along the first principal component (horizontal axis) hold a correlation of **-0.76** with the length of the sentences. This means that having a sentences to the right of the PCA diagram tend of have smaller lengths. It is also worthwhile ot note that the clusters have similar shapes across both classes in train and test explaining that the learned embeddings have decent generalization performances.



(a) Train: Real news

(b) Train: Satire

(c) Test: Real news

(d) Test: Satire

(e) Visualization of sentence embeddings of satire news using PCA

Figure 3: Visualization of sentence embeddings using PCA

We show the PCA decomposition for a satire news article containing 15 sentences (Table 4) in Figure 3e. We clearly see that sentences 2, 3, 4, 6, 7, 10, and 13 (lying in the upper cluster) are separated from the rest of the sentences (lying in the central cluster). The high correlation between the horizontal axis and the length of the sentence is also visible in Figure 3e. It is also interesting to note that sentences 8, 5, 1, 15 in the lower cluster are more verbose as opposed to the upper cluster samples 10, 2, 6 indicating the learned embeddings of the S module could capture verbosity as a fine-grained property of satire.

4.4.3 Analysis of Attention Layer

We normalized the attention weights (using "Min-Max" normalization) of the Attention layer of Model B, such that maximum weight in a news article is 1, and the minimum is 0. Firstly, we observe that the attention weights of the <PAD> sentences are less than 0.01 times the mean attention weight of the other sentences of the news article. This shows the effectiveness of the Attention layer in ignoring <PAD> sentences, thus not adding their contribution in the final document embedding. Next, we observe that the weight (normalized) of the last sentence of a news article is high for satirical news, while is almost 0 for real news. Figure 4a shows the distribution of the attention weights (normalized) over a news article, for a 3 satirical news and 3 real news from the test data. The mean normalized attention weight of the last sentence of satirical news in **0.728**, while the same for real news is **0.07**. Since these weights directly contribute to the document embedding (which we have shown to be able to distinguish satire effectively), they must reflect the amount of satire present in the sentence. Thus, having a higher weight would mean the sentence is more relevant to capture satire. This means that the last sentence of a new article contains

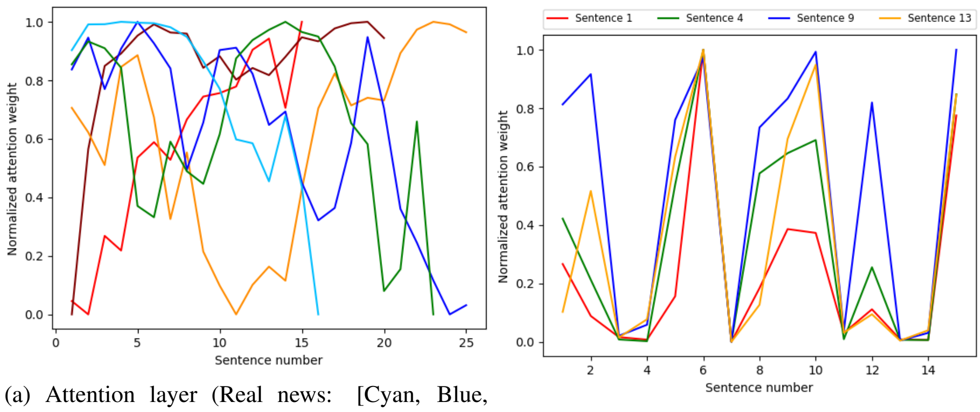
No.	Sentence	Att.
1	OTTAWA With the holiday season drawing near , Prime Minister Harper issued a statement today urging all Canadians to enjoy the time of year that brings us closer to our most beloved industries	0.045
2	“ I think that Canadians understand what this season is really about , ” the Prime Minister said . ”	0.0
3	We ’ ve all seen enough Holiday movies to know that what really matters is unfeelingly capitalizing on the emotions of others in order to make millions upon millions of dollars . ”	0.268
4	“ Actually , now that I mention it , that is what every single other season of the year is also about . ”	0.218
5	The statement , released along with a Christmas card featuring the members of the Prime Minister ’ s estranged family standing with blank faces in front of a Sears , has already roused the nation into a frenzy of holiday cheer .	0.535
6	“ Giving is fine , ” said local man Derek Williams , backhand slapping an elderly woman away from a pile of One Direction themed Furbies .	0.588
7	“ But what really matters is the opportunity to reward multinationals for doing business in my country . ”	0.528
8	Since the statement was released , retail outlets and superstores have made record amounts of money : money which will allow them to make more money , which in turn will be put toward the noble cause of making yet more money.	0.666
9	The Prime Minister has expressed pleasure at the fact that even the innocuous act of giving a gift to a loved one has been subjected to the economics of reckless consumerism .	0.744
10	“ Look at all of these Legos , ” said the Prime Minister , gesturing at the pile of new boxes of Lego covering Laureen ’ s otherwise empty half of the bed .	0.756
11	“ I was given these Legos .	0.778
12	Now , of all the people in my family , I have the most Legos .	0.904
13	Economically speaking , I am the best . ”	0.942
14	This is not the first time the Prime Minister has used holidays to spread cheer to everyday Canadians .	0.705
15	Last Valentine ’ s day , Mr Harper urged Canadians to spice up their romantic lives by switching their personal lubricant from boring old petroleum jelly to titillating new bitumen sand jelly .	1.0

Table 4: Normalized attention weights (from the Attention Layer of Model B) of a satire news article

more satirical features if the news is satirical. Therefore, we conclude that the last sentence of a news article is a key feature for detecting satire.

4.4.4 Analysis of Attentive-Concatenate Layer

The normalized attention weights of the Attentive-Concatenate layer reveal that, in the context of each of the sentences of a news article, there are a few key sentences that contain relevant satire information. Figure 4b show the distribution of attention weights (normalized) of the Attentive-Concatenate layer for a few sentences of the news article shown in Table 4. We find that for each sentence, the attention distribution peaks at the same set of sentences (more or less), across all news articles. From this, we can draw the conclusion that these sentences must be more important to capture satire than the other sentences present in the news. For the news article in Table 4, these key sentences are sentence 6, 10, 12 and 15. We also note that the final sentence is one of those key sentences that are important to detect satire. Thus, we conclude that news satire is effectively decided by a few key sentences of the news article.



(a) Attention layer (Real news: [Cyan, Blue, Green], Satire news: [Red, Maroon, Orange])

(b) Attentive-Concatenate Layer

Figure 4: Normalized attention weights

5 Conclusion

We proposed a novel, robust, plug-and-play hierarchical architecture for detecting subtle linguistic nuances like satire in news. Our approach achieves comparable results with the existing state-of-the-art models, without the use of any hand-crafted linguistic feature reflecting satire. This hierarchical approach enables to learn satire features both at the sentence level (learned by S module) and at the document level (learned by D module). An extensive comparison with several state-of-the-art methods for satire news detection was also explored on a real world satire news dataset. We experimented with different choices of initial word embeddings and different S and D modules that include CNN, LSTM and GRU. Experimental results showed slightly superior performance of our proposed architecture with the combination CNN as the S module and a Attentive Deep Bidirectional GRU (3 layers deep) as the D module, using Glove vectors concatenated with syntactic information as the input embeddings to be performing the best. The architecture apart from state-of-the-art detection performances, allowed us to perform fine-grained sentence level analyses giving us a deeper insight into the phenomena of satire. An analysis of the learned models revealed the existence of a few key sentences (including the last sentence) that are important to detect satire. For our future work, we wish to explore Recursive neural networks in order to incorporate the structure of the language for better modeling of sentences.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments. This work was supported in part by the NSF grant 1527364.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Clint Burfoot and Timothy Baldwin. 2009. Automatic satire detection: Are you having a laugh? In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pages 161–164. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Niall J Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116. Association for Computational Linguistics.

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

Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *AAAI*, pages 2972–2978.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Arjun Mukherjee, 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.

Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 632–640. ACM.

Shereen Oraby, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn A Walker, and Steve Whittaker. 2015. And that's a fact: Distinguishing factual and emotional argumentation in online dialogue. In *ArgMining@ HLT-NAACL*, pages 116–126.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics.

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

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

Victoria L Rubin, Niall J Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of NAACL-HLT*, pages 7–17.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news. *arXiv preprint arXiv:1703.06959*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical news detection and analysis using attention mechanism and linguistic features. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hu Zhang, Zhuohua Fan, Jia-heng Zheng, and Quanming Liu. 2012. An improving deception detection method in computer-mediated communication. *JNW*, 7(11):1811–1816.