Opinion Mining with Deep Recurrent Neural Networks

Ozan İrsoy and **Claire Cardie**
Department of Computer Science
Cornell University
Ithaca, NY, 14853, USA
`oirsoy, cardie@cs.cornell.edu`

Abstract

Recurrent neural networks (RNNs) are connectionist models of sequential data that are naturally applicable to the analysis of natural language. Recently, "depth in space" — as an orthogonal notion to "depth in time" — in RNNs has been investigated by stacking multiple layers of RNNs and shown empirically to bring a temporal hierarchy to the architecture. In this work we apply these *deep RNNs* to the task of opinion expression extraction formulated as a token-level sequence-labeling task. Experimental results show that deep, narrow RNNs outperform traditional shallow, wide RNNs with the same number of parameters. Furthermore, our approach outperforms previous CRF-based baselines, including the state-of-the-art semi-Markov CRF model, and does so without access to the powerful opinion lexicons and syntactic features relied upon by the semi-CRF, as well as without the standard layer-by-layer pre-training typically required of RNN architectures.

1 Introduction

Fine-grained opinion analysis aims to detect the subjective expressions in a text (e.g. "hate") and to characterize their intensity (e.g. strong) and sentiment (e.g. negative) as well as to identify the opinion holder (the entity expressing the opinion) and the target, or topic, of the opinion (i.e. what the opinion is about) (Wiebe et al., 2005). Fine-grained opinion analysis is important for a variety of NLP tasks including opinion-oriented question answering and opinion summarization. As a result, it has been studied extensively in recent years.

In this work, we focus on the detection of opinion expressions — both *direct subjective expressions* (DSEs) and *expressive subjective expressions* (ESEs) as defined in Wiebe et al. (2005). DSEs consist of explicit mentions of private states or speech events expressing private states; and ESEs consist of expressions that indicate sentiment, emotion, etc., without explicitly conveying them. An example sentence shown in Table 1 in which the DSE "has refused to make any statements" explicitly expresses an opinion holder's attitude and the ESE "as usual" indirectly expresses the attitude of the writer.

The	committee	,	as	usual	,	has
O	O	O	B_ESE	I_ESE	O	B_DSE
refused	to	make	any	statements	.	
I_DSE	I_DSE	I_DSE	I_DSE	I_DSE	O	

Table 1: An example sentence with labels

Opinion extraction has often been tackled as a sequence labeling problem in previous work (e.g. Choi et al. (2005)). This approach views a sentence as a sequence of tokens labeled using the conventional BIO tagging scheme: B indicates the beginning of an opinion-related expression, I is used for tokens inside the opinion-related expression, and O indicates tokens outside any opinion-related class. The example sentence in Table 1 shows the appropriate tags in the BIO scheme. For instance, the ESE "as usual" results in the tags B_ESE for "as" and I_ESE for "usual".

Variants of **conditional random field (CRF)** approaches have been successfully applied to opinion expression extraction using this token-based view (Choi et al., 2005; Breck et al., 2007): the state-of-the-art approach is the semiCRF, which relaxes the Markovian assumption inherent to CRFs and operates at the phrase level rather than the token level, allowing the incorporation of phrase-level features (Yang and Cardie, 2012). The success of the CRF- and semiCRF-based approaches, however, hinges critically on access to an appropriate feature set, typically based on constituent and dependency parse trees, manually crafted opinion lexicons, named entity taggers and other preprocessing components (see Yang and Cardie (2012) for an up-to-date list).

Distributed representation learners provide a different approach to learning in which latent features are modeled as distributed dense vectors of hidden layers. A **recurrent neural network (RNN)** is one such learner that can operate on sequential data of variable length, which means it can also be applied as a sequence labeler. Moreover, **bidirectional RNNs** incorporate information from preceding as well as following tokens (Schuster and Paliwal, 1997) while recent advances in word embedding induction (Collobert and Weston, 2008; Mnih and Hinton, 2007; Mikolov et al., 2013; Turian et al., 2010) have enabled more effective training of RNNs by allowing a lower dimensional dense input representation and hence, more compact networks (Mikolov et al., 2010; Mesnil et al., 2013). Finally, **deep recurrent networks**, a type of RNN with multiple stacked hidden layers, are shown to naturally employ a temporal hierarchy with multiple layers operating at different time scales (Hermans and Schrauwen, 2013): lower levels capture short term interactions among words; higher layers reflect interpretations aggregated over longer spans of text. When applied to natural language sentences, such hierarchies might better model the multi-scale language effects that are emblematic of natural languages, as suggested by previous results (Hermans and Schrauwen, 2013).

Motivated by the recent success of deep architectures in general and deep recurrent networks in particular, we explore an application of **deep bidirectional RNNs** — henceforth **deep RNNs** — to the task of opinion expression extraction. For both DSE and ESE detection, we show that such models outperform conventional, shallow (uni- and bidirectional) RNNs as well as previous CRF-based state-of-the-art baselines, including the semiCRF model.

In the rest of the paper we discuss related work (Section 2) and describe the architecture and training methods for recurrent neural networks (RNNs), bidirectional RNNs, and deep (bidirectional) RNNs (Section 3). We present experiments using a standard corpus for fine-grained opinion extraction in Section 4.

2 Related Work

Opinion extraction. Early work on fine-grained opinion extraction focused on recognizing subjective phrases (Wilson et al., 2005; Munson et al., 2005). Breck et al. (2007), for example, formulated the problem as a token-level sequence-labeling problem and apply a CRF-based approach, which significantly outperformed previous baselines. Choi et al. (2005) extended the sequential prediction approach to jointly identify opinion holders; Choi and Cardie (2010) jointly detected polarity and intensity along with the opinion expression. Reranking approaches have also been explored to improve the performance of a single sequence labeler (Johansson and Moschitti, 2010; Johansson and Moschitti, 2011). More recent work relaxes the Markovian assumption of CRFs to capture phrase-level interactions, significantly improving upon the token-level labeling approach (Yang and Cardie, 2012). In particular, Yang and Cardie (2013) propose a joint inference model to jointly detect opinion expressions, opinion holders and targets, as well as the relations among them, outperforming previous pipelined approaches.

Deep learning. Recurrent neural networks (Elman, 1990) constitute one important class of naturally deep architecture that has been applied to many sequential prediction tasks. In the context of NLP, recurrent neural networks view a sentence as a sequence of tokens and have been successfully applied to tasks such as language modeling (Mikolov et al., 2011) and spoken language understanding (Mesnil et al., 2013). Since classical recurrent neural networks only incorporate information from the past (i.e. preceding tokens), **bidirectional** variants have been proposed to incorporate information from both the past and the future (i.e. subsequent tokens) (Schuster and Paliwal, 1997). Bidirectionality is especially useful for NLP tasks, since information provided by the following tokens is generally helpful (and sometimes essential) when making a decision on the current token.

Stacked recurrent neural networks have been proposed as a way of constructing *deep* RNNs (Schmidhuber, 1992; El Hihi and Bengio, 1995). Careful empirical investigation of this architecture showed that multiple layers in the stack can operate at different time scales (Hermans and Schrauwen, 2013). Pascanu et al. (2013) explore other ways of constructing deep RNNs that are orthogonal to the concept of stacking layers on top of each other. In this work, we focus on the *stacking* notion of depth.

3 Methodology

This section describes the architecture and training methods for the deep bidirectional recurrent networks that we propose for the task of opinion expression mining. Recurrent neural networks are presented in 3.1, bidirectionality is introduced in 3.2, and deep bidirectional RNNs, in 3.3.

3.1 Recurrent Neural Networks

A recurrent neural network (Elman, 1990) is a class of neural network that has recurrent connections, which allow a form of memory. This makes them applicable for sequential prediction tasks with arbitrary spatio-temporal dimensions. Thus, their structure fits many NLP tasks, when the interpretation of a single sentence is viewed as analyzing a sequence of tokens. In this work, we focus our attention on only Elman-type networks (Elman, 1990).

In an **Elman-type network**, the hidden layer h_t at time step t is computed from a nonlinear transformation of the current input layer x_t and the previous hidden layer h_{t-1}. Then, the final output y_t is computed using the hidden layer h_t. One can interpret h_t as an intermediate representation summarizing the past, which is used to make a final decision on the current input.

More formally, given a sequence of vectors $\{x_t\}_{t=1..T}$, an Elman-type RNN operates by computing the following memory and output sequences:

$$h_t = f(Wx_t + Vh_{t-1} + b) \tag{1}$$

$$y_t = g(Uh_t + c) \tag{2}$$

where f is a nonlinear function, such as the sigmoid function and g is the output nonlinearity, such as the

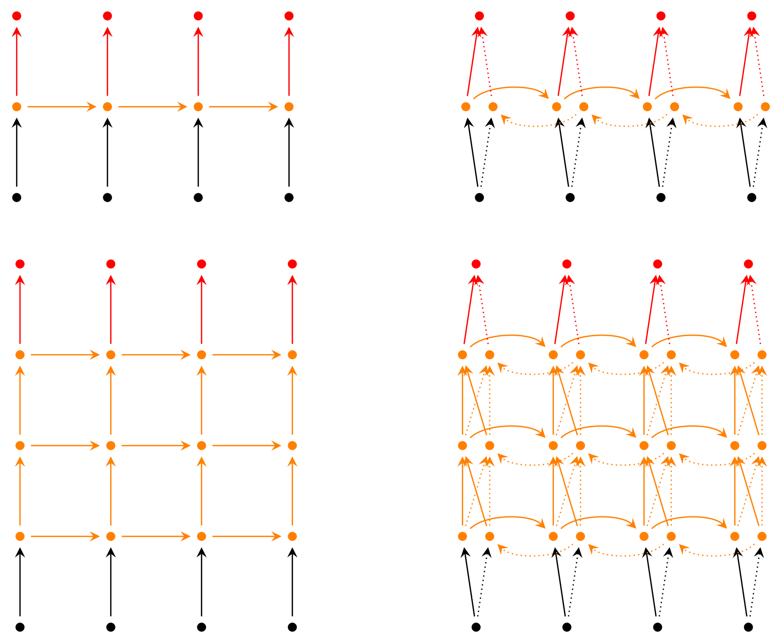

Figure 1: Recurrent neural networks. Each black, orange and red node denotes an input, hidden or output layer, respectively. Solid and dotted lines denote the connections of forward and backward layers, respectively. Top: Shallow unidirectional (left) and bidirectional (right) RNN. Bottom: 3-layer deep unidirectional (left) and bidirectional (right) RNN.

softmax function. W and V are weight matrices between the input and hidden layer, and among the hidden units themselves (connecting the previous intermediate representation to the current one), respectively, while U is the output weight matrix. b and c are bias vectors connected to hidden and output units, respectively. As a base case for the recursion in Equation 1, h_0 is assumed to be 0.

Training an RNN can be done by optimizing a discriminative objective (e.g. the cross entropy for classification tasks) with a gradient-based method. Backpropagation through time can be used to efficiently compute the gradients (Werbos, 1990). This method is essentially equivalent to unfolding the network in time and using backpropagation as in feedforward neural networks, while sharing the connection weights across different time steps. The Elman-style RNN is shown in Figure 1, top left.

3.2 Bidirectionality

Observe that with the above definition of RNNs, we have information only about the past, when making a decision on x_t. This is limiting for most NLP tasks. As a simple example, consider the two sentences: “I did not accept his suggestion” and “I did not go to the rodeo”. The first has a DSE phrase (“did not accept”) and the second does not. However, any such RNN will assign the same labels for the words “did” and “not” in both sentences, since the preceding sequences (past) are the same: the Elman-style unidirectional RNNs lack the representational power to model this task. A simple way to work around this problem is to include a fixed-size future context around a single input vector (token). However, this approach requires tuning the context size, and ignores future information from outside of the context window. Another way to incorporate information about the future is to add bidirectionality to the architecture, referred as the **bidirectional RNN** (Schuster and Paliwal, 1997):

$$\overrightarrow{h}_t = f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + \overrightarrow{b}) \quad (3)$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b}) \quad (4)$$

$$y_t = g(U_{\rightarrow}\overrightarrow{h}_t + U_{\leftarrow}\overleftarrow{h}_t + c) \quad (5)$$

where $\overrightarrow{W}$, $\overrightarrow{V}$ and $\overrightarrow{b}$ are the forward weight matrices and bias vector as before; $\overleftarrow{W}$, $\overleftarrow{V}$ and $\overleftarrow{b}$ are their backward counterparts; $U_{\rightarrow}$, $U_{\leftarrow}$ are the output matrices; and c is the output bias.[1] Again, we assume $\overrightarrow{h}_0 = \overleftarrow{h}_{T+1} = 0$. In this setting $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ can be interpreted as a summary of the past, and the future, respectively, around the time step t. When we make a decision on an input vector, we employ the two intermediate representations $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ of the past and

[1]As a convention, we adopt the following notation throughout the paper: Superscript arrows for vectors disambiguate between forward and backward representations. Superscript arrows for matrices denote the resulting vector representations (connection outputs), and subscript arrows for matrices denote incoming vector representations (connection inputs). We omit subscripts when there is no ambiguity.

the future. (See Figure 1, top right.) Therefore in the bidirectional case, we have perfect information about the sequence (ignoring the practical difficulties about capturing long term dependencies, caused by vanishing gradients), whereas the classical Elman-type network uses only partial information as described above.

Note that the forward and backward parts of the network are independent of each other until the output layer when they are combined. This means that during training, after backpropagating the error terms from the output layer to the forward and backward hidden layers, the two parts can be thought of as separate, and each trained with the classical backpropagation through time (Werbos, 1990).

3.3 Depth in Space

Recurrent neural networks are often characterized as having *depth in time*: when unfolded, they are equivalent to feedforward neural networks with as many hidden layers as the number tokens in the input sequence (with shared connections across multiple layers of time). However, this notion of depth likely does not involve hierarchical processing of the data: across different time steps, we repeatedly apply the same transformation to compute the memory contribution of the input (W), to compute the response value from the current memory (U) and to compute the next memory vector from the previous one (V). Therefore, assuming the input vectors $\{x_t\}$ together lie in the same representation space, as do the output vectors $\{y_t\}$, hidden representations $\{h_t\}$ lie in the same space as well. As a result, they do not necessarily become more and more abstract, hierarchical representations of one another as we traverse in time. However in the more conventional, *stacked* deep learners (e.g. deep feedforward nets), an important benefit of depth is the hierarchy among hidden representations: every hidden layer conceptually lies in a different representation space, and constitutes a more abstract and higher-level representation of the input (Bengio, 2009).

In order to address these concerns, we investigate deep RNNs, which are constructed by stacking Elman-type RNNs on top of each other (Hermans and Schrauwen, 2013). Intuitively, every layer of the deep RNN treats the memory sequence of the previous layer as the input sequence, and computes its own memory representation.

More formally, we have:

$$\begin{aligned}\overrightarrow{h}_t^{(i)} &= f(\overrightarrow{W}_{\rightarrow}^{(i)} \overrightarrow{h}_t^{(i-1)} + \overrightarrow{W}_{\leftarrow}^{(i)} \overleftarrow{h}_t^{(i-1)} \\ &\quad + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)}) \qquad (6)\\ \overleftarrow{h}_t^{(i)} &= f(\overleftarrow{W}_{\rightarrow}^{(i)} \overrightarrow{h}_t^{(i-1)} + \overleftarrow{W}_{\leftarrow}^{(i)} \overleftarrow{h}_t^{(i-1)} \\ &\quad + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)}) \qquad (7)\end{aligned}$$

when $i > 1$ and

$$\overrightarrow{h}_t^{(1)} = f(\overrightarrow{W}^{(1)} x_t + \overrightarrow{V}^{(1)} \overrightarrow{h}_{t-1}^{(1)} + \overrightarrow{b}^{(1)}) \qquad (8)$$

$$\overleftarrow{h}_t^{(1)} = f(\overleftarrow{W}^{(1)} x_t + \overleftarrow{V}^{(1)} \overleftarrow{h}_{t+1}^{(1)} + \overleftarrow{b}^{(1)}) \qquad (9)$$

Importantly, note that both forward and backward representations are employed when computing the forward and backward memory of the next layer.

Two alternatives for the output layer computations are to employ all memory layers or only the last. In this work we adopt the second approach:

$$y_t = g(U_{\rightarrow} \overrightarrow{h}_t^{(L)} + U_{\leftarrow} \overleftarrow{h}_t^{(L)} + c) \qquad (10)$$

where L is the number of layers. Intuitively, connecting the output layer to only the last hidden layer forces the architecture to capture enough high-level information at the final layer for producing the appropriate output-layer decision.

Training a deep RNN can be conceptualized as interleaved applications of the conventional backpropagation across multiple layers, and backpropagation through time within a single layer.

The unidirectional and bidirectional deep RNNs are depicted in the bottom half of Figure 1.

Hypotheses. In general, we expected that the deep RNNs would show the most improvement over shallow RNNS for ESEs — phrases that implicitly convey subjectivity. Existing research has shown that these are harder to identify than direct expressions of subjectivity (DSEs): they are variable in length and involve terms that, in many (or most) contexts, are neutral with respect to sentiment and subjectivity. As a result, models that do a better job interpreting the context should be better at disambiguating subjective vs. non-subjective uses of phrases involving common words (e.g. "as usual", "in fact"). Whether or not deep RNNs would be powerful enough to outperform the state-of-the-art semiCRF was unclear, especially if the semiCRF is given access to the distributed word representations (embeddings) employed by the deep RNNs. In addition, the semiCRF has access to parse tree information and opinion lexicons, neither of which is available to the deep RNNs.

4 Experiments

Activation Units. We employ the standard softmax activation for the output layer: $g(x) = e^{x_i} / \sum_j e^{x_j}$. For the hidden layers we use the rectifier linear activation: $f(x) = \max\{0, x\}$. Experimentally, rectifier activation gives better performance, faster convergence, and sparse representations. Previous work also reported good results when training deep neural networks using rectifiers, without a pretraining step (Glorot et al., 2011).

Data. We use the MPQA 1.2 corpus (Wiebe et al., 2005) (535 news articles, 11,111 sentences) that is manually annotated with both DSEs and ESEs at the phrase level. As in previous work, we separate 135 documents as a development set and employ 10-fold CV over the remaining 400 documents. The development set is used during cross validation to do model selection.

Layers	$\|h\|$	Precision		Recall		F1	
		Prop.	Bin.	Prop.	Bin.	Prop	Bin.
Shallow	36	62.24	65.90	65.63*	73.89*	63.83	69.62
Deep 2	29	63.85*	67.23*	65.70*	**74.23***	64.70*	70.52*
Deep 3	25	63.53*	67.67*	65.95*	73.87*	64.57*	70.55*
Deep 4	22	**64.19***	**68.05***	**66.01***	73.76*	**64.96***	**70.69***
Deep 5	21	60.65	61.67	56.83	69.01	58.60	65.06
Shallow	200	62.78	66.28	65.66*	74.00*	64.09	69.85
Deep 2	125	62.92*	66.71*	66.45*	**74.70***	64.47	70.36
Deep 3	100	**65.56***	**69.12***	**66.73***	74.69*	**66.01***	**71.72***
Deep 4	86	61.76	65.64	63.52	72.88*	62.56	69.01
Deep 5	77	61.64	64.90	62.37	72.10	61.93	68.25

Table 2: Experimental evaluation of RNNs for DSE extraction

Layers	$\|h\|$	Precision		Recall		F1	
		Prop.	Bin.	Prop.	Bin.	Prop	Bin.
Shallow	36	51.34	59.54	57.60	72.89*	54.22	65.44
Deep 2	29	51.13	59.94	**61.20***	**75.37***	**55.63***	**66.64***
Deep 3	25	**53.14***	**61.46***	58.01	72.50	55.40*	66.36*
Deep 4	22	51.48	60.59*	59.25*	73.22	54.94	66.15*
Deep 5	21	49.67	58.42	48.98	65.36	49.25	61.61
Shallow	200	**52.20***	60.42*	58.11	72.64	54.75	65.75
Deep 2	125	51.75*	60.75*	60.69*	74.39*	55.77*	66.79*
Deep 3	100	52.04*	**60.50***	**61.71***	**76.02***	**56.26***	**67.18***
Deep 4	86	50.62*	58.41*	53.55	69.99	51.98	63.60
Deep 5	77	49.90*	57.82	52.37	69.13	51.01	62.89

Table 3: Experimental evaluation of RNNs for ESE extraction

Evaluation Metrics. We use precision, recall and F-measure for performance evaluation. Since the boundaries of expressions are hard to define even for human annotators (Wiebe et al., 2005), we use two soft notions of the measures: *Binary Overlap* counts every overlapping match between a predicted and true expression as correct (Breck et al., 2007; Yang and Cardie, 2012), and *Proportional Overlap* imparts a partial correctness, proportional to the overlapping amount, to each match (Johansson and Moschitti, 2010; Yang and Cardie, 2012). All statistical comparisons are done using a two-sided paired t-test with a confidence level of $\alpha = .05$.

Baselines (CRF and SEMICRF). As baselines, we use the CRF-based method of Breck et al. (2007) and the SEMICRF-based method of Yang and Cardie (2012), which is the state-of-the-art in opinion expression extraction. Features that the baselines use are words, part-of-speech tags and membership in a manually constructed opinion lexicon (within a [-1, +1] context window). Since SEMICRF relaxes the Markovian assumption and operates at the segment-level instead of the token-level, it also has access to parse trees of sentences to generate candidate segments (Yang and Cardie, 2012).

Word Vectors (+VEC). We also include versions of the baselines that have access to pre-trained word vectors. In particular, CRF+VEC employs word vectors as continuous features per every token. Since SEMICRF has phrase-level rather than word-level features, we simply take the mean of every word vector for a phrase-level vector representation for SEMICRF+VEC as suggested in Mikolov et al. (2013).

In all of our experiments, we keep the word vectors fixed (i.e. do not finetune) to reduce the degree of freedom of our models. We use the publicly available 300-dimensional word vectors of Mikolov et al. (2013), trained on part of the Google News dataset (~100B words). Preliminary experiments with other word vector representations such as Collobert-Weston (2008) embeddings or HLBL (Mnih and Hinton, 2007) provided poorer results ($\sim -3\%$ difference in proportional and binary F1).

Regularizer. We do not employ any regularization for smaller networks (~24,000 parameters) because we have not observed strong overfitting (i.e. the difference between training and test performance is small). Larger networks are regularized with the recently proposed dropout technique (Hinton et al., 2012): we randomly set entries of hidden representations to 0 with a probability called the dropout rate, which is tuned over the development set. Dropout prevents learned

	Model		Precision		Recall		F1	
			Prop.	Bin.	Prop.	Bin.	Prop	Bin.
DSE	CRF		74.96*	82.28*	46.98	52.99	57.74	64.45
	semiCRF		61.67	69.41	**67.22***	73.08*	64.27	71.15*
	CRF	+vec	**74.97***	**82.43***	49.47	55.67	59.59	66.44
	semiCRF	+vec	66.00	71.98	60.96	68.13	63.30	69.91
	Deep RNN	3 100	65.56	69.12	66.73*	**74.69***	**66.01***	**71.72***
ESE	CRF		56.08	68.36	42.26	51.84	48.10	58.85
	semiCRF		45.64	69.06	58.05	64.15	50.95	66.37*
	CRF	+vec	**57.15***	69.84*	44.67	54.38	50.01	61.01
	semiCRF	+vec	53.76	**70.82***	52.72	61.59	53.10	65.73
	Deep RNN	3 100	52.04	60.50	**61.71***	**76.02***	**56.26***	**67.18***

Table 4: Comparison of Deep RNNs to state-of-the-art (semi)CRF baselines for DSE and ESE detection

features from co-adapting, and it has been reported to yield good results when training deep neural networks (Krizhevsky et al., 2012; Dahl et al., 2013).

Network Training. We use the standard multiclass cross-entropy as the objective function when training the neural networks. We use stochastic gradient descent with momentum with a fixed learning rate (.005) and a fixed momentum rate (.7). We update weights after minibatches of 80 sentences. We run 200 epochs for training. Weights are initialized from small random uniform noise. We experiment with networks of various sizes, however we have the same number of hidden units across multiple forward and backward hidden layers of a single RNN. We do not employ a pre-training step; deep architectures are trained with the supervised error signal, even though the output layer is connected to only the final hidden layer. With these configurations, every architecture successfully converges without any oscillatory behavior. Additionally, we employ early stopping for the neural networks: out of all iterations, the model with the best development set performance (Proportional F1) is selected as the final model to be evaluated.

4.1 Results and Discussion

Bidirectional vs. Unidirectional. Although our focus is on bidirectional RNNs, we first confirm that the SHALLOW bidirectional RNN outperforms a (shallow) unidirectional RNN for both DSE and ESE recognition. To make the comparison fair, each network has the same number of total parameters: we use 65 hidden units for the unidirectional, and 36 for the bidirectional network, respectively. Results are as expected: the bidirectional RNN obtains higher F1 scores than the unidirectional RNN — 63.83 vs. 60.35 (proportional overlap) and 69.62 vs. 68.31 (binary overlap) for DSEs; 54.22 vs. 51.51 (proportional) and 65.44 vs. 63.65 (binary) for ESEs. All differences are statistically significant at the 0.05 level. Thus, we will not include comparisons to the unidirectional RNNs in the remaining experiments.

Adding Depth. Next, we quantitatively investigate the effects of adding depth to RNNs. Tables 2 and 3 show the evaluation of RNNs of various depths and sizes. In both tables, the first group networks have approximately 24,000 parameters and the second group networks have approximately 200,000 parameters. Since all RNNs within a group have approximately the same number of parameters, they grow narrower as they get deeper. Within each group, bold shows the best result with an asterisk denoting statistically indistinguishable performance with respect to the best. As noted above, all statistical comparisons use a two-sided paired t-test with a confidence level of $\alpha = .05$.

In both DSE and ESE detection and for larger networks (bottom set of results), 3-layer RNNs provide the best results. For smaller networks (top set of results), 2, 3 and 4-layer RNNs show equally good performance for certain sizes and metrics and, in general, adding additional layers degrades performance. This could be related to how we train the architectures as well as to the decrease in width of the networks. In general, we observe a trend of increasing performance as we increase the number of layers, until a certain depth.

deepRNNs vs. (semi)CRF. Table 4 shows comparison of the best deep RNNs to the previous best results in the literature. In terms of F-measure, DEEP RNN performs best for both DSE and ESE detection, achieving a new state-of-the-art performance for the more strict proportional overlap measure, which is harder to improve upon than the binary evaluation metric. SEMICRF, with its very high recall, performs comparably to the DEEP RNN on the binary metric. Note that RNNs do not have access to any features other than word vectors.

In general, CRFs exhibit high precision but low recall (CRFs have the best precision on both DSE and ESE detection) while SEMICRFs exhibit a high recall, low precision performance. Compared to SEMICRF, the DEEP RNNs produce an even higher recall but sometimes lower precision for ESE detection. This suggests that the methods are complementary, and can

(1)	
	The situation obviously remains fluid from hour to hour but it [seems to be] [going in the right direction]
DEEPRNN	The situation [obviously] remains fluid from hour to hour but it [seems to be going in the right] direction
SHALLOW	The situation [obviously] remains fluid from hour to hour but it [seems to be going in] the right direction
SEMICRF	The situation [obviously remains fluid from hour to hour but it seems to be going in the right direction]
(2)	
	have always said this is a multi-faceted campaign [but equally] we have also said any future military action [would have to be based on evidence] , ...
DEEPRNN	have always said this is a multi-faceted campaign but [equally we] have also said any future military action [would have to be based on evidence] , ...
SHALLOW	have always said this is a multi-faceted [campaign but equally we] have also said any future military action would have to be based on evidence , ...
SEMICRF	have always said this is a multi-faceted campaign but equally we have also said any future military action would have to be based on evidence , ...
(3)	
	Ruud Lubbers , the United Nations Commissioner for Refugees , said Afghanistan was [not yet] secure for aid agencies to operate in and " [not enough] " food had been taken into the country .
DEEPRNN	Ruud Lubbers , the United Nations Commissioner for Refugees , said Afghanistan was [not yet] secure for aid agencies to operate in and " [not enough] " food had been taken into the country .
SHALLOW	Ruud Lubbers , the United Nations Commissioner for Refugees , said Afghanistan was [not yet] secure for aid agencies to operate in and " [not enough] " food had been taken into the country .
SEMICRF	Ruud Lubbers , the United Nations Commissioner for Refugees , said Afghanistan was not yet secure for aid agencies to operate in and " not enough " food had been taken into the country .

Figure 2: Examples of output. In each set, the gold-standard annotations are shown in the first line.

potentially be even more powerful when combined in an ensemble method.

Word vectors. Word vectors help CRFs on both precision and recall on both tasks. However, SEMICRFs become more conservative with word vectors, producing higher precision and lower recall on both tasks. This sometimes hurts overall F-measure.

Among the (SEMI)CRF-based methods, SEMICRF obtains the highest F1 score for DSEs and for ESEs using the softer metric; SEMICRF+VEC performs best for ESEs according to the stricter proportional overlap measure.

Network size. Finally, we observe that even small networks (such as 4-layer deep RNN for DSE and 2-layer deep RNN for ESE) outperform conventional CRFs. This suggests that with the help of good word vectors, we can train compact but powerful sequential neural models.

When examining the output, we see some systematic differences between the previously top-performing SEMICRF and the RNN-based models. (See Figure 2.) First, SEMICRF often identifies excessively long subjective phrases as in Example 1. Here, none of the models exactly matches the gold standard, but the RNNs are much closer. And all three models appear to have identified an ESE that was mistakenly omitted by the human annotator — "obviously". At the same time, the SEMICRF sometimes entirely misses subjective expressions that the RNNs identify — this seems to occur when there are no clear indications of sentiment in the subjective expression. The latter can be seen in Examples 2 and 3, in which the SEMICRF does not identify "but equally", "would have to be based on evidence", "not yet", and "not enough".

We also observe evidence of the power of the DEEPRNN over the SHALLOWRNN in Examples 4 and 5. (See Figure 3.) In contrast to Figure 2, Figure 3 distinguishes subjective expressions that are (correctly) assigned an initial Begin label from those that consist only of Inside labels[2] — the latter are shown in ALL CAPS and indicate some degree of confusion in the model that produced them. In Example 4, SHALLOWRNN exhibits *some* evidence for each ESE — it labels one or more tokens as Inside an ESE ("any" and "time"). But it does not explicitly tag the beginning of the ESE. DEEPRNN does better, identifying the first ESE in its entirety ("in any case") and identifying more words as being Inside the second ESE ("it is high time). A similar situation occurs in Example 5.

5 Conclusion

In this paper we have explored an application of deep recurrent neural networks to the task of sentence-level opinion expression extraction. We empirically evaluated deep RNNs against conventional, shallow RNNs that have only a single hidden layer. We also compared our models with previous (semi)CRF-based approaches.

Experiments showed that deep RNNs outperformed shallow RNNs on both DSE and ESE extrac-

[2]Sequences of I's are decoded as the associated DSE or ESE even though they lack the initial B.

(4)

	[In any case] , [it is high time] that a social debate be organized ...
DEEPRNN	[In any case] , it is HIGH TIME that a social debate be organized ...
SHALLOW	In ANY case , it is high TIME that a social debate be organized ...

(5)

	Mr. Stoiber [has come a long way] from his refusal to [sacrifice himself] for the CDU in an election that [once looked impossible to win] , through his statement that he would [under no circumstances] run against the wishes...
DEEPRNN	Mr. Stoiber [has come a long way from] his [refusal to sacrifice himself] for the CDU in an election that [once looked impossible to win] , through his statement that he would [under no circumstances run against] the wishes...
SHALLOW	Mr. Stoiber has come A LONG WAY FROM his refusal to sacrifice himself for the CDU in an election that [once looked impossible] to win , through his statement that he would under NO CIRCUMSTANCES run against the wishes...

Figure 3: DEEPRNN Output vs. SHALLOWRNN Output. In each set of examples, the gold-standard annotations are shown in the first line. Tokens assigned a label of Inside with no preceding Begin tag are shown in ALL CAPS.

tion. Furthermore, deep RNNs outperformed previous (semi)CRF baselines, achieving new state-of-the-art results for fine-grained on opinion expression extraction.

We have trained our deep networks without any pre-training and with only the last hidden layer connected to the output layer. One potential future direction is to explore the effects of pre-training on the architecture. Pre-training might help to exploit the additional representational power available in deeper networks. Another direction is to investigate the impact of fine-tuning the word vectors during supervised training. Additionally, alternative notions of depth that are orthogonal to stacking, as in Pascanu et al. (2013) can be investigated for this task.

Acknowledgments

This work was supported in part by NSF grant IIS-1314778 and DARPA DEFT Grant FA8750-13-2-0015. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, DARPA or the U.S. Government.

References

Yoshua Bengio. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI*, pages 2683–2688.

Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 269–274. Association for Computational Linguistics.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of HLT/EMNLP*, pages 355–362. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

George E Dahl, Tara N Sainath, and Geoffrey E Hinton. 2013. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE.

Salah El Hihi and Yoshua Bengio. 1995. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in Neural Information Processing Systems*, pages 493–499.

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

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, volume 15, pages 315–323.

Michiel Hermans and Benjamin Schrauwen. 2013. Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 190–198.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Lan-*

guage Learning, pages 67–76. Association for Computational Linguistics.

Richard Johansson and Alessandro Moschitti. 2011. Extracting opinion expressions and their polarities: exploration of pipelines and joint models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 101–106. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, volume 1, page 4.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. Interspeech.

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

Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

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.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM.

M Arthur Munson, Claire Cardie, and Rich Caruana. 2005. Optimizing to arbitrary nlp metrics using ensemble selection. In *Proceedings of HLT/EMNLP*, pages 539–546. Association for Computational Linguistics.

Razvan Pascanu, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2013. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Jürgen Schmidhuber. 1992. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242.

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

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP*, pages 347–354. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1335–1345. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of ACL*.