Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings

Pengfei Liu[1], Shafiq Joty[2] and Helen Meng[1]

[1]The Chinese University of Hong Kong,
Hong Kong SAR of China
[2]Qatar Computing Research Institute,
Doha, Qatar

{pfliu, hmmeng}@se.cuhk.edu.hk, sjoty@qf.org.qa

September 20, 2015

The Chinese University of Hong Kong 香港中文大學 系統工程與工程管理學系
Department of Systems Engineering & Engineering Management

Outline

Outline

Fine-grained Opinion Mining Tasks

$\underbrace{\textbf{John}}_{\text{opinion holder}}$ says, the $\underbrace{\textbf{hard disk}}_{\text{opinion target}}$ is $\underbrace{\textbf{very noisy}}_{\text{opinion expression}}$

Fine-grained Opinion Mining [Wiebe et al., 2005]

- identifying the opinion holder
- identifying the target/aspect of the opinion
- detecting opinion expressions
- measuring strength/polarity of opinion expressions

Fine-grained Opinion Mining Tasks

Token-level Sequence Labeling

- identifying the opinion holder
- identifying the opinion target
- detecting opinion expressions

The	**hard**	**disk**	is	*very*	*noisy*
O	B-TARG	I-TARG	O	O	O
O	O	O	O	B-EXPR	I-EXPR

Semantic Compositional Task [Socher et al., 2013]

- measure strength/polarity of opinion expressions
- compose word vectors based on parse trees

Our Objective: General class of models to solve these tasks

Fine-grained Opinion Mining Tasks

Token-level Sequence Labeling

- identifying the opinion holder
- identifying the opinion target
- detecting opinion expressions

The	**hard**	**disk**	is	*very*	*noisy*
O	B-TARG	I-TARG	O	O	O
O	O	O	O	B-EXPR	I-EXPR

Semantic Compositional Task [Socher et al., 2013]

- measure strength/polarity of opinion expressions
- compose word vectors based on parse trees

Our Objective: General class of models to solve these tasks

Contributions

(1) Propose a general class of discriminative models based on **Recurrent Neural Network** (RNN) architecture and **word embeddings** for fine-grained opinion mining tasks

(2) Experiment with several RNN architectures including Elman-type, Jordan-type and Long Short Term Memory (**LSTM**) and their variations

(3) Present a new architecture to incorporate other **linguistic features** into RNNs besides word embeddings

Contributions

(1) Propose a general class of discriminative models based on **Recurrent Neural Network** (RNN) architecture and **word embeddings** for fine-grained opinion mining tasks

(2) Experiment with several RNN architectures including Elman-type, Jordan-type and Long Short Term Memory (**LSTM**) and their variations

(3) Present a new architecture to incorporate other **linguistic features** into RNNs besides word embeddings

Contributions

(1) Propose a general class of discriminative models based on **Recurrent Neural Network** (RNN) architecture and **word embeddings** for fine-grained opinion mining tasks

(2) Experiment with several RNN architectures including Elman-type, Jordan-type and Long Short Term Memory (**LSTM**) and their variations

(3) Present a new architecture to incorporate other **linguistic features** into RNNs besides word embeddings

Outline

Motivations for Recurrent Neural Networks (RNNs)

(1) **Extensive feature engineering efforts** required by other models (e.g., CRFs) for each task [Pontiki et al., 2014]

(2) DNNs **learn features automatically** and **outperform** CRFs on similar tasks

(3) **Word embeddings** yield significant gains when used as extra features in existing NLP systems [Turian et al., 2010]

(4) Word embeddings also help in **effective training** of RNNs [Collobert and Weston, 2008, Irsoy and Cardie, 2014]

We apply RNNs to

- Model sequential dependencies
- Learn features automatically
- Incorporate linguistic features into RNNs

Motivations for Recurrent Neural Networks (RNNs)

(1) **Extensive feature engineering efforts** required by other models (e.g., CRFs) for each task [Pontiki et al., 2014]

(2) DNNs **learn features automatically** and **outperform** CRFs on similar tasks

(3) **Word embeddings** yield significant gains when used as extra features in existing NLP systems [Turian et al., 2010]

(4) Word embeddings also help in **effective training** of RNNs [Collobert and Weston, 2008, Irsoy and Cardie, 2014]

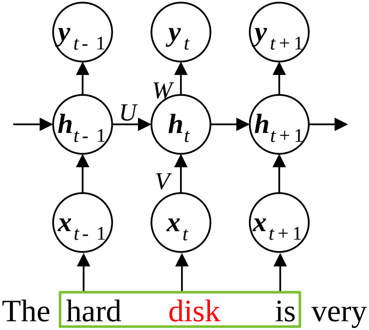
We apply RNNs to

- Model sequential dependencies
- Learn features automatically
- Incorporate linguistic features into RNNs

RNN Types

- Elman-RNN [Elman, 1990]
- Jordan-RNN [Jordan, 1997]
- LSTM-RNN [Hochreiter and Schmidhuber, 1997]

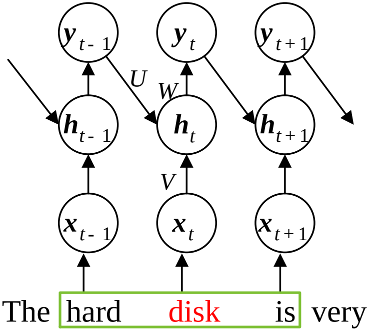
Elman-type RNN



$$\mathbf{h}_t = f(U\mathbf{h}_{t-1} + V\mathbf{x}_t + \mathbf{b}) \tag{1}$$

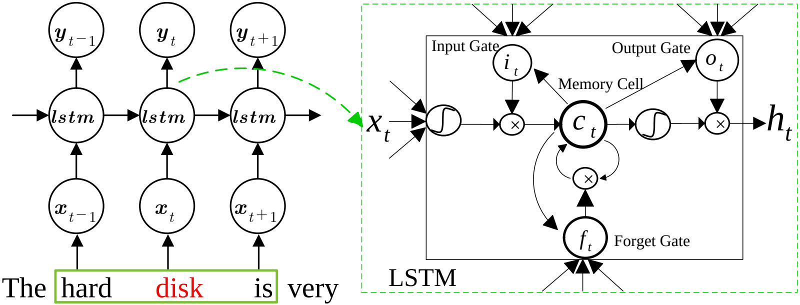
- Concatenated context vector for "disk": $x_t = [x_{hard}, x_{disk}, x_{is}]$

Jordan-type RNN



$$\mathbf{h}_t = f(U\mathbf{y}_{t-1} + V\mathbf{x}_t + \mathbf{b}) \tag{2}$$

LSTM-RNN



$$\mathbf{i}_t = \sigma(U_i\mathbf{h}_{t-1} + V_i\mathbf{x}_t + C_i\mathbf{c}_{t-1} + \mathbf{b}_i) \quad (3)$$

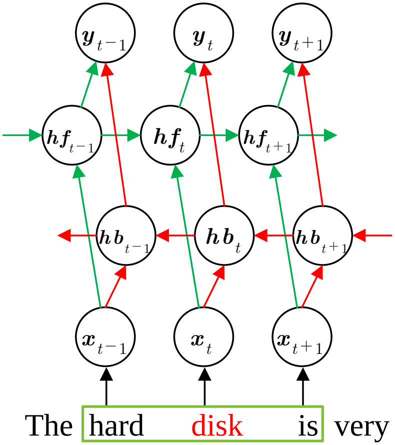
$$\mathbf{f}_t = \sigma(U_f\mathbf{h}_{t-1} + V_f\mathbf{x}_t + C_f\mathbf{c}_{t-1} + \mathbf{b}_f) \quad (4)$$

$$\mathbf{c}_t = \mathbf{i}_t \odot g(U_c\mathbf{h}_{t-1} + V_c\mathbf{x}_t + \mathbf{b}_c) + \mathbf{f}_t \odot \mathbf{c}_{t-1} \quad (5)$$

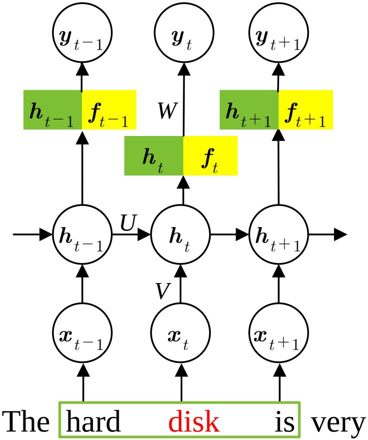
$$\mathbf{o}_t = \sigma(U_o\mathbf{h}_{t-1} + V_o\mathbf{x}_t + C_o\mathbf{c}_t + \mathbf{b}_o) \quad (6)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot h(\mathbf{c}_t) \quad (7)$$

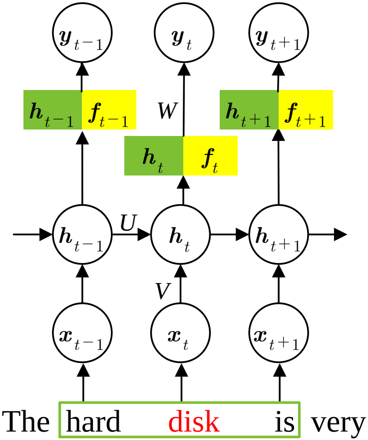
Bidirectionality



Linguistic Features



- Encode POS and chunk information as **binary features**
- Feed them to the **output layer** of RNNs



- Encode POS and chunk information as **binary features**
- Feed them to the **output layer** of RNNs

Outline

Word Embeddings (1/2)

Word Embeddings

- Distributed, real-valued and dense representation
- Each dimension describes **syntactic/semantic** properties
- Induced using **neural networks** from very large corpora

Fine-tuning

- Random weight initialization $\Rightarrow$ local minima
- Embeddings as features $\Rightarrow$ not exploiting feature learning
- Our model $\Rightarrow$ **fine-tuning** of pre-trained embeddings

	SENNA[1]	Google[2]	Amazon[3]
Domain	Wikipedia	News	Reviews
Vocabulary	130K	3M	1M
#Words	N/A	100B	4.7B
#Dimensions	50	300	50 & 300

- Feed to CRF as additional **continuous valued features**
- Initialize the **lookup-table layer** of a RNN

[1] [Collobert and Weston, 2008]
[2] [Mikolov et al., 2013]
[3] [McAuley and Leskovec, 2013]

Outline

Datasets & Evaluation Metrics

	Laptop		Restaurant	
	Train	Test	Train	Test
Sentences	3045	800	3041	800
Sentence length	15	13	14	14
One-word targets	1494	364	2786	818
Multi-word targets	864	290	907	316
Total targets	2358	654	3693	1134

- Precision, Recall and F_1 based on **exact matching**

Baseline

Second-order linear-chain CRF:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}, t)\right) \tag{8}$$

- $\mathbf{x} = \{x_t\}_{t=1}^{T}$: word sequence from $t = 1$ to T
- $\mathbf{y} = \{y_t\}_{t=1}^{T}$: label sequence of *B, I or O*
- θ_k: weight parameter for feature function $f_k(y_t, y_{t-1}, \mathbf{x}, t)$
- K: the total number of feature functions
- $Z(\mathbf{x})$: normalizing factor over all label sequences

Features for CRF Baseline: POS, chunks, prefixes, suffixes, position, context, stylistics

Experiments on Train/Test Set (CRF vs. RNNs)

System	Dim.	$\|h_l\|$	Laptop	$\|h_r\|$	Restaurant
CRF Base	-	-	68.66	-	77.28
+SENNA	50	-	71.38	-	78.54
+Amazon	50	-	70.61	-	79.46
+Google	300	-	68.81	-	**80.36**
+Amazon	300	-	**72.20**	-	79.66
Jordan-RNN					
+SENNA	50	200	71.41	200	78.83
+Amazon	50	100	73.21	150	79.01
+Google	300	150	**73.42**	200	**79.89**
+Amazon	300	50	72.43	200	78.30
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82

- Word embeddings complement other features for CRF
- RNNs outperform CRF without any feature engineering
- Elman-RNN generally performs better than Jordan-RNN

Experiments on Train/Test Set (CRF vs. RNNs)

System	Dim.	$\lvert h_l \rvert$	Laptop	$\lvert h_r \rvert$	Restaurant
CRF Base	-	-	68.66	-	77.28
+SENNA	50	-	71.38	-	78.54
+Amazon	50	-	70.61	-	79.46
+Google	300	-	68.81	-	**80.36**
+Amazon	300	-	**72.20**	-	79.66
Jordan-RNN					
+SENNA	50	200	71.41	200	78.83
+Amazon	50	100	73.21	150	79.01
+Google	300	150	**73.42**	200	**79.89**
+Amazon	300	50	72.43	200	78.30
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82

- Word embeddings complement other features for CRF
- RNNs outperform CRF without any feature engineering
- Elman-RNN generally performs better than Jordan-RNN

Experiments on Train/Test Set (CRF vs. RNNs)

System	Dim.	$\|h_l\|$	Laptop	$\|h_r\|$	Restaurant
CRF Base	-	-	68.66	-	77.28
+SENNA	50	-	71.38	-	78.54
+Amazon	50	-	70.61	-	79.46
+Google	300	-	68.81	-	**80.36**
+Amazon	300	-	**72.20**	-	79.66
Jordan-RNN					
+SENNA	50	200	71.41	200	78.83
+Amazon	50	100	73.21	150	79.01
+Google	300	150	**73.42**	200	**79.89**
+Amazon	300	50	72.43	200	78.30
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82

- Word embeddings complement other features for CRF
- RNNs outperform CRF without any feature engineering
- Elman-RNN generally performs better than Jordan-RNN

Experiments on Train/Test Set (Bidirection & LSTM)

System	Dim.	$\lvert h_l \rvert$	Laptop	$\lvert h_r \rvert$	Restaurant
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82
Bi-Elman-RNN					
+SENNA	50	100	72.38	100	**80.10**
+Amazon	50	50	**73.93**	50	79.97
+Google	300	50	72.67	100	79.52
+Amazon	300	50	71.12	50	79.09
LSTM-RNN					
+SENNA	50	100	73.40	150	79.43
+Amazon	50	50	72.44	50	**79.79**
+Google	300	100	72.11	50	79.20
+Amazon	300	50	**73.52**	50	78.99
Bi-LSTM-RNN					
+SENNA	50	50	72.60	150	**79.89**
+Amazon	50	100	**74.03**	100	79.36
+Google	300	50	70.90	50	78.80
+Amazon	300	150	71.25	150	78.88

- Bidirection and LSTM provide no clear gains over Elman-RNN
- LSTM and Bidirection increase the number of parameters
  $\Rightarrow$ may contribute to overfitting on this specific task

Experiments on Train/Test Set (Bidirection & LSTM)

System	Dim.	$\lvert h_l \rvert$	Laptop	$\lvert h_r \rvert$	Restaurant
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82
Bi-Elman-RNN					
+SENNA	50	100	72.38	100	**80.10**
+Amazon	50	50	**73.93**	50	79.97
+Google	300	50	72.67	100	79.52
+Amazon	300	50	71.12	50	79.09
LSTM-RNN					
+SENNA	50	100	73.40	150	79.43
+Amazon	50	50	72.44	50	**79.79**
+Google	300	100	72.11	50	79.20
+Amazon	300	50	**73.52**	50	78.99
Bi-LSTM-RNN					
+SENNA	50	50	72.60	150	**79.89**
+Amazon	50	100	**74.03**	100	79.36
+Google	300	50	70.90	50	78.80
+Amazon	300	150	71.25	150	78.88

- Bidirection and LSTM provide no clear gains over Elman-RNN
- LSTM and Bidirection increase the number of parameters
  $\Rightarrow$ may contribute to overfitting on this specific task

Experiments on Train/Test Set (Linguistic Features)

System	Dim.	$\|h_l\|$	Laptop	$\|h_r\|$	Restaurant
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82
Elman-RNN + Feat.					
+SENNA	50	50	73.70	100	81.36
+Amazon	50	200	73.30	50	**81.66**
+Google	300	150	**74.25**	100	80.57
+Amazon	300	50	73.92	100	80.24
LSTM-RNN					
+SENNA	50	100	73.40	150	79.43
+Amazon	50	50	72.44	50	**79.79**
+Google	300	100	72.11	50	79.20
+Amazon	300	50	**73.52**	50	78.99
LSTM-RNN + Feat.					
+SENNA	50	50	73.19	150	80.28
+Amazon	50	100	**75.00**	50	80.82
+Google	300	50	72.19	50	**81.37**
+Amazon	300	100	72.85	100	80.60
SemEval-14 top systems					
IHS_RD	-	-	**74.55**	-	79.62
DLIREC	-	-	73.78	-	**84.01**

- Linguistic features yield gains on both datasets
- RNNs without feature engineering achieve the second best
- **LSTM-RNN+Feat.** achieves the best results on Laptop

Experiments on Train/Test Set (Linguistic Features)

System	Dim.	$\lvert h_l \rvert$	Laptop	$\lvert h_r \rvert$	Restaurant
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82
Elman-RNN + Feat.					
+SENNA	50	50	73.70	100	81.36
+Amazon	50	200	73.30	50	**81.66**
+Google	300	150	**74.25**	100	80.57
+Amazon	300	50	73.92	100	80.24
LSTM-RNN					
+SENNA	50	100	73.40	150	79.43
+Amazon	50	50	72.44	50	**79.79**
+Google	300	100	72.11	50	79.20
+Amazon	300	50	**73.52**	50	78.99
LSTM-RNN + Feat.					
+SENNA	50	50	73.19	150	80.28
+Amazon	50	100	**75.00**	50	80.82
+Google	300	50	72.19	50	**81.37**
+Amazon	300	100	72.85	100	80.60
SemEval-14 top systems					
IHS_RD	-	-	**74.55**	-	79.62
DLIREC	-	-	73.78	-	**84.01**

- Linguistic features yield gains on both datasets
- RNNs without feature engineering achieve the second best
- **LSTM-RNN+Feat.** achieves the best results on Laptop

Experiments on Train/Test Set (Linguistic Features)

System	Dim.	$\|h_l\|$	Laptop	$\|h_r\|$	Restaurant
Elman-RNN					
+SENNA	50	100	73.86	150	79.89
+Amazon	50	100	**74.43**	100	**80.37**
+Google	300	100	72.91	100	79.54
+Amazon	300	200	73.67	100	79.82
Elman-RNN + Feat.					
+SENNA	50	50	73.70	100	81.36
+Amazon	50	200	73.30	50	**81.66**
+Google	300	150	**74.25**	100	80.57
+Amazon	300	50	73.92	100	80.24
LSTM-RNN					
+SENNA	50	100	73.40	150	79.43
+Amazon	50	50	72.44	50	**79.79**
+Google	300	100	72.11	50	79.20
+Amazon	300	50	**73.52**	50	78.99
LSTM-RNN + Feat.					
+SENNA	50	50	73.19	150	80.28
+Amazon	50	100	**75.00**	50	80.82
+Google	300	50	72.19	50	**81.37**
+Amazon	300	100	72.85	100	80.60
SemEval-14 top systems					
IHS_RD	-	-	**74.55**	-	79.62
DLIREC	-	-	73.78	-	**84.01**

- Linguistic features yield gains on both datasets
- RNNs without feature engineering achieve the second best
- **LSTM-RNN+Feat.** achieves the best results on Laptop

10-fold Cross Validation Results

Model	Laptop			Restaurant		
	P	R	F_1	P	R	F_1
CRF Base	**79.77**	64.09	70.87	**82.59**	74.63	78.36
+ SENNA	78.23	**67.38**	**72.34**	81.21	**78.12**	**79.60**
Elman-RNN	**82.03**	72.68	76.97	81.96	78.41	80.08
+ Feat.	80.02	**76.60**	**78.22**	81.91	**81.22**	81.52
+ Bidir.	81.92	73.70	77.47	81.69	78.46	79.97
+ Feat. + Bidir.	81.00	75.70	78.17	**82.80**	80.44	**81.57**
LSTM-RNN	**81.92**	73.30	77.14	**83.64**	77.45	80.36
+ Feat.	80.70	**75.82**	**78.00**	81.80	**81.39**	81.54
+ Bidir.	81.31	74.20	77.37	81.66	79.23	80.37
+ Feat. + Bidir.	80.81	74.48	77.27	82.96	80.42	**81.56**

- Linguistic features **complement** word embeddings in RNNs
  - Laptop: +1.25% ($p < 0.004$)
  - Restaurant: +1.44% ($p < 0.00006$)
- Elman/LSTM + Feat. obtains best results on Laptop
- Elman/LSTM + Feat. + Bidir. obtains best results on Restaurant

10-fold Cross Validation Results

Model	Laptop			Restaurant		
	P	R	F_1	P	R	F_1
CRF Base	**79.77**	64.09	70.87	**82.59**	74.63	78.36
+ SENNA	78.23	**67.38**	**72.34**	81.21	**78.12**	**79.60**
Elman-RNN	**82.03**	72.68	76.97	81.96	78.41	80.08
+ Feat.	80.02	**76.60**	**78.22**	81.91	**81.22**	81.52
+ Bidir.	81.92	73.70	77.47	81.69	78.46	79.97
+ Feat. + Bidir.	81.00	75.70	78.17	**82.80**	80.44	**81.57**
LSTM-RNN	**81.92**	73.30	77.14	**83.64**	77.45	80.36
+ Feat.	80.70	**75.82**	**78.00**	81.80	**81.39**	81.54
+ Bidir.	81.31	74.20	77.37	81.66	79.23	80.37
+ Feat. + Bidir.	80.81	74.48	77.27	82.96	80.42	**81.56**

- Linguistic features **complement** word embeddings in RNNs
  - Laptop: +1.25% ($p < 0.004$)
  - Restaurant: +1.44% ($p < 0.00006$)
- Elman/LSTM + Feat. obtains best results on Laptop
- Elman/LSTM + Feat. + Bidir. obtains best results on Restaurant

10-fold Cross Validation Results

Model	Laptop			Restaurant		
	P	R	F_1	P	R	F_1
CRF Base	**79.77**	64.09	70.87	**82.59**	74.63	78.36
+ SENNA	78.23	**67.38**	**72.34**	81.21	**78.12**	**79.60**
Elman-RNN	**82.03**	72.68	76.97	81.96	78.41	80.08
+ Feat.	80.02	**76.60**	**78.22**	81.91	**81.22**	81.52
+ Bidir.	81.92	73.70	77.47	81.69	78.46	79.97
+ Feat. + Bidir.	81.00	75.70	78.17	**82.80**	80.44	**81.57**
LSTM-RNN	**81.92**	73.30	77.14	**83.64**	77.45	80.36
+ Feat.	80.70	**75.82**	**78.00**	81.80	**81.39**	81.54
+ Bidir.	81.31	74.20	77.37	81.66	79.23	80.37
+ Feat. + Bidir.	80.81	74.48	77.27	82.96	80.42	**81.56**

- Linguistic features **complement** word embeddings in RNNs
  - Laptop: +1.25% ($p < 0.004$)
  - Restaurant: +1.44% ($p < 0.00006$)
- Elman/LSTM + Feat. obtains best results on Laptop
- Elman/LSTM + Feat. + Bidir. obtains best results on Restaurant

Effects of Fine-tuning

System	Dim.	Laptop		Restaurant	
Elman-RNN		-tune	+tune	-tune	+tune
+SENNA	50	60.85	73.86	75.78	79.89
+Amazon	50	15.51	**74.43**	22.85	**80.37**
+Random	50	38.26	72.99	56.98	78.44
+Google	300	67.91	72.91	74.73	79.54
+Amazon	300	15.51	73.67	22.85	79.82
Jordan-RNN		-tune	+tune	-tune	+tune
+SENNA	50	58.81	71.41	74.68	78.83
+Amazon	50	15.51	73.21	22.85	79.01
+Random	50	38.05	71.46	55.65	77.38
+Google	300	69.39	**73.42**	77.33	**79.89**
+Amazon	300	15.51	72.43	22.85	78.30

- In most cases **fine-tuning** makes a big difference!

Outline

Conclusions

General class of models for fine-grained opinion mining

- Pre-trained word embeddings from three external sources
- RNNs including Elman, Jordan, LSTM and their variations

Results on extracting opinion targets

- Word embeddings improve both CRF and RNN models
- RNNs outperform CRFs
- Incorporating linguistic features into RNNs improves further
- Fine-tuning word embeddings gives the best results

Conclusions

General class of models for fine-grained opinion mining

- Pre-trained word embeddings from three external sources
- RNNs including Elman, Jordan, LSTM and their variations

Results on extracting opinion targets

- Word embeddings improve both CRF and RNN models
- RNNs outperform CRFs
- Incorporating linguistic features into RNNs improves further
- Fine-tuning word embeddings gives the best results

Questions

Thank You!

References I

Collobert, R. and Weston, J. (2008).
A unified architecture for natural language processing: deep neural networks with multitask learning.
In *Proceedings of ICML*, pages 160–167. ACM.

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

Hochreiter, S. and Schmidhuber, J. (1997).
Long short-term memory.
Neural Comput., 9(8):1735–1780.

Irsoy, O. and Cardie, C. (2014).
Opinion mining with deep recurrent neural networks.
In *Proceedings of EMNLP*, pages 720–728.

Jordan, M. I. (1997).
Serial order: A parallel distributed processing approach.
Advances in psychology, 121:471–495.

McAuley, J. and Leskovec, J. (2013).
Hidden factors and hidden topics: understanding rating dimensions with review text.
In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.
In *Advances in Neural Information Processing Systems*, pages 3111–3119.

References II

Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., and Manandhar, S. (2014).
Semeval-2014 task 4: Aspect based sentiment analysis.
In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013).
Recursive deep models for semantic compositionality over a sentiment treebank.
In *Proceedings of EMNLP*, pages 1631–1642. Citeseer.

Turian, J., Ratinov, L., and Bengio, Y. (2010).
Word representations: a simple and general method for semi-supervised learning.
In *Proceedings of the 48th Annual Meeting of ACL*, pages 384–394. ACL.

Wiebe, J., Wilson, T., and Cardie, C. (2005).
Annotating expressions of opinions and emotions in language.
Language resources and evaluation, 39(2-3):165–210.

Appendix: Features for CRF Baseline

- **Character features**: 'AllUpper', 'AllDigit', 'AllSymbol', 'AllUpperDigit', 'AllUpperSymbol', 'AllDigitSymbol', 'AllUpperDigitSymbol', 'InitUpper', 'AllLetter', 'AllAlnum', two digits, four digits, all alphanumerical, not alphanumerical, containing an upper-or-lower-or-digit-or-symbol character;
- **BOS feature** for begin of sentence and **EOS feature** for end of sentence;
- **Context features** by combining the above features from context (within left two and right two). For example, we generate a bigram feature of "w[0]|w[1]=I|charge", which means the current word is I and the next word is charge, or a bigram feature of "pos[0]|pos[1]=PRP|VBP" which means the current POS tag is PRP and the next POS tag is VBP.

Linguistic Features

Feature	POS tags	Comment
JJ	JJ, JJR, JJS	Adjective
NN	NN, NNS, NNP, NNPS	Noun
RB	RB, RBR, RBS	Adverb
VB	VB, VBD, VBG, VBN, VBP, VBZ	Verb

Table 1: Features for POS tags

Feature	Chunking tags	Phrase
NP	B-NP, I-NP	Noun
PP	B-PP, I-PP	Prepositional
VP	B-VP, I-VP	Verb
ADJP	B-ADJP, I-ADJP	Adjective
ADVP	B-ADVP, I-ADVP	Adverb

Table 2: Features for Chunking

- Encode POS tags and chunk as **binary features**
- Feed them to the **output layer** of RNNs