Using Structured Events to Predict Stock Price Movement: An Empirical Investigation

Xiao Ding[†,*], Yue Zhang[‡], Ting Liu[†], Junwen Duan[†]
[†]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{xding, tliu, jwduan}@ir.hit.edu.cn
[‡]Singapore University of Technology and Design
yue_zhang@sutd.edu.sg

Summary

- What
  - News events predict stock trend
- Why
  - Previous works model events based on shallow features
    - Bag-of-words, named entities and noun phrases
- How
  - Model events based on structured entity-relation information
    - Open Information Extraction
  - Use deep neural networks to predict

Apple has sued Samsung Electronics for copying 'the look and feel' of its iPad tablet and iPhone smartphone.
=>
{"Apple", "sued", "Samsung", "Electronics", "copying", ...}

Method

- Event representation
  - E = (O1, P, O2, T)
  - Actor O1 performs action P on object O2 at timestamp T
- Event extraction
  - Event phrase extraction
    - Syntactic constraint
    - Lexical constraint
  - Argument extraction
- Event generalization
  - WordNet
  - VerbNet

Sep 3, 2013 - Microsoft agrees to buy Nokia’s mobile phone business for $7.2 billion.
=>
(Actor = Microsoft, Action = buy, Object = Nokia’s mobile phone business, Time = Sep 3, 2013)

Instant view: Private sector adds 114,000 jobs in July: ADP.
=>
(Private sector, adds, 114,000 jobs)

(Private sector, adds, 114,000 jobs)
=>
(private sector, multiply_class, 114,000 job)

Method (Cont.)

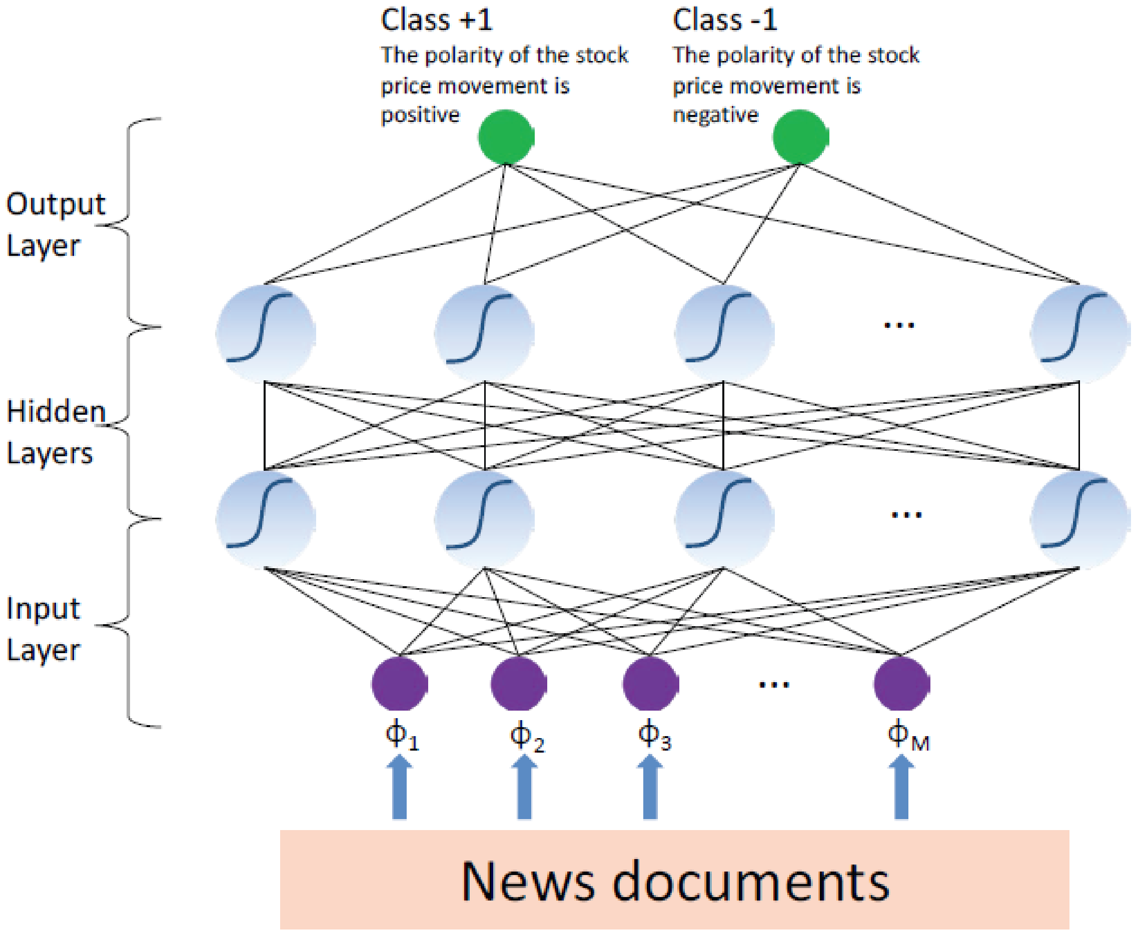
- Deep Neural Network Model

$$y_{cls} = f(net_{cls}) = \sigma(\boldsymbol{w_{cls}} \cdot \mathbf{y_2}) \qquad (1)$$

$$\begin{aligned} y_{2k} &= \sigma(\boldsymbol{w_{2k}} \cdot \mathbf{y_1}) \quad (k \in [1,\ |y_2|]) \\ y_{1j} &= \sigma(\boldsymbol{w_{1j}} \cdot \boldsymbol{\Phi}(d_n)) \quad (j \in [1,\ |y_1|]) \end{aligned} \qquad (2)$$

(O1, P, O2, T)
=>
(O1, P, O2, O1 + P, P + O2, O1 + P + O2)

(Microsoft, buy, Nokia's mobile phone business)
=>
(#arg1=Microsoft, #action=get class, #arg2=Nokia's mobile phone business, #arg1 action=Microsoft get class, #action arg2=get class Nokia's mobile phone business, #arg1 action arg2=Microsoft get class Nokia's mobile phone business)



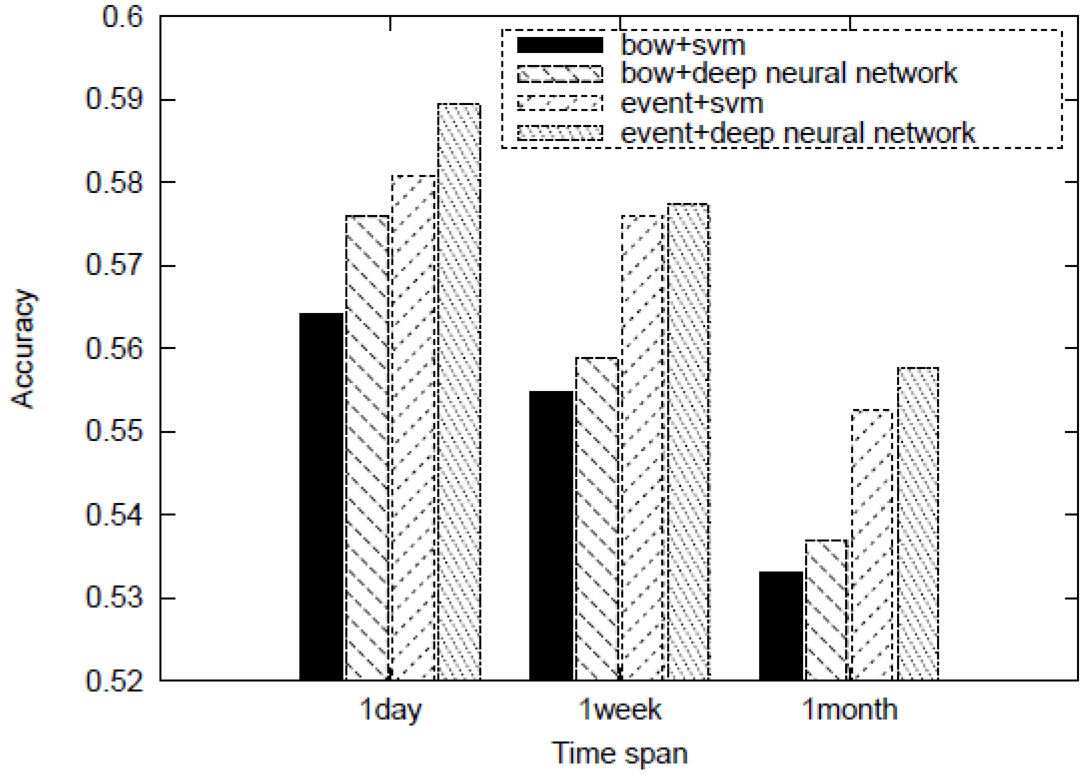
Experiments

	train	dev	test
number of instances	1425	178	179
number of events	54776	6457	6593
time interval	02/10/2006 - 18/16/2012	19/06/2012 - 21/02/2013	22/02/2013 - 21/11/2013

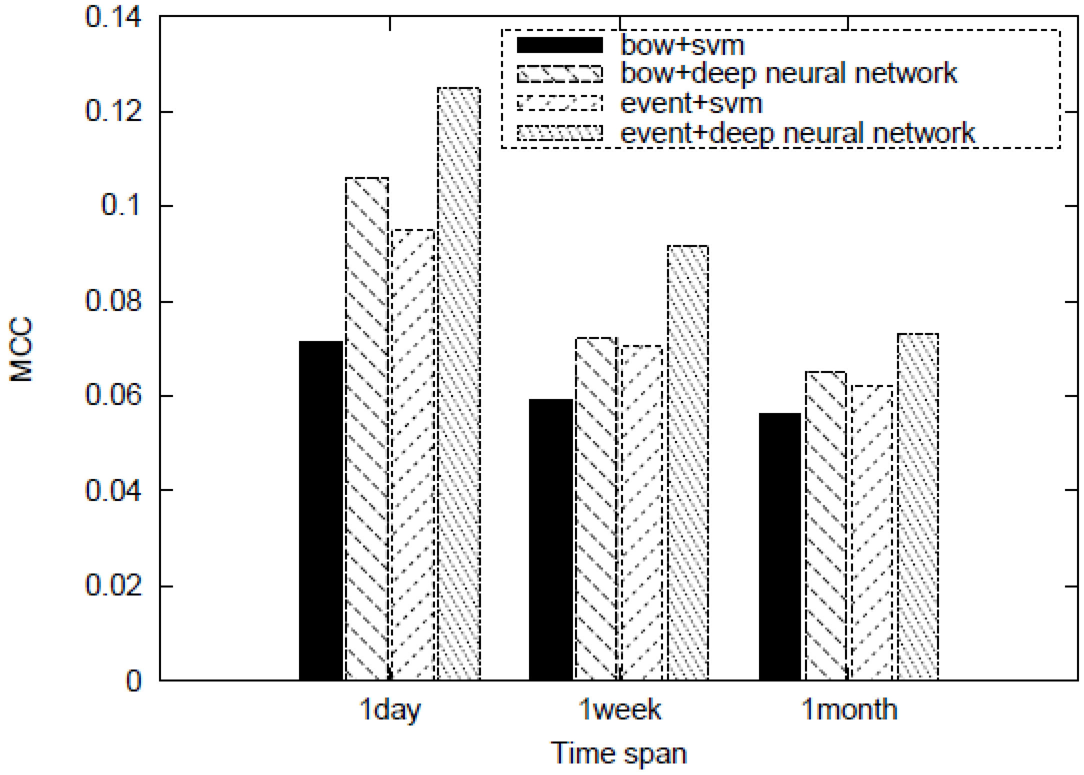
- Data
  - Reuters and Bloomberg
  - S&P 500, 15 individual shares
- Metrics
  - Accuracy
  - MCC

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (3)$$

Experiments (Cont.)



(a) Accuarcy



(b) MCC

Figure 3: Overall development experiment results

Experiments (Cont.)

		1 day	1 week	1 month
1 layer	Accuracy	58.94%	57.73%	55.76%
	MCC	0.1249	0.0916	0.0731
2 layers	Accuracy	59.60%	57.73%	56.19%
	MCC	0.1683	0.1215	0.0875

Table 2: Different numbers of hidden layers

	title	content	content + title	bloomberg title + title
Acc	59.60%	54.65%	56.83%	59.64%
MCC	0.1683	0.0627	0.0852	0.1758

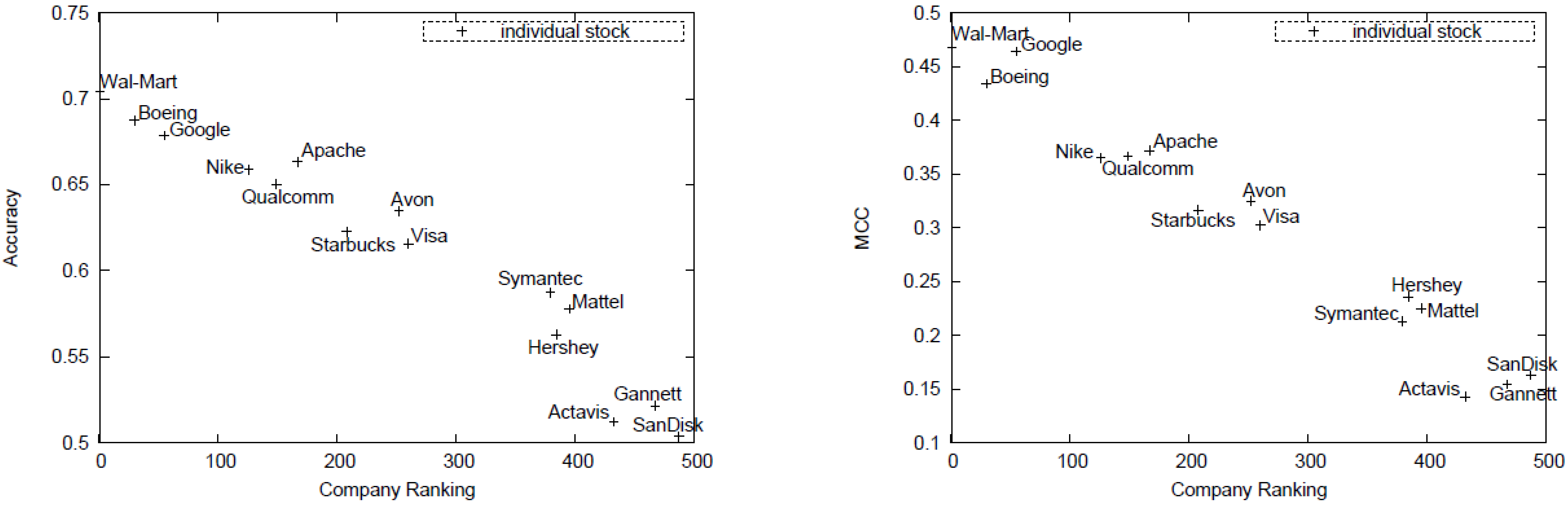
Table 3: Different amounts of data

Experiments (Cont.)

Google Inc.					
Company News		Sector News		All News	
Acc	MCC	Acc	MCC	Acc	MCC
67.86%	0.4642	61.17%	0.2301	55.70%	0.1135
Boeing Company					
Company News		Sector News		All News	
Acc	MCC	Acc	MCC	Acc	MCC
68.75%	0.4339	57.14%	0.1585	56.04%	0.1605
Wal-Mart Stores					
Company News		Sector News		All News	
Acc	MCC	Acc	MCC	Acc	MCC
70.45%	0.4679	62.03%	0.2703	56.04%	0.1605

Table 4: Individual stock prediction results

Experiments (Cont.)



(a) Accuarcy

(b) MCC

Figure 4: Individual stock prediction experiment results

Experiments (Cont.)

	S&P 500 Index Prediction		Individual Stock Prediction					
			Google Inc.		Boeing Company		Wal-Mart Stores	
	Accuracy	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy	MCC
dev	59.60%	0.1683	67.86%	0.4642	68.75%	0.4339	70.45%	0.4679
test	58.94%	0.1649	66.97%	0.4435	68.03%	0.4018	69.87%	0.4456

Table 5: Final experimental results on the test dataset

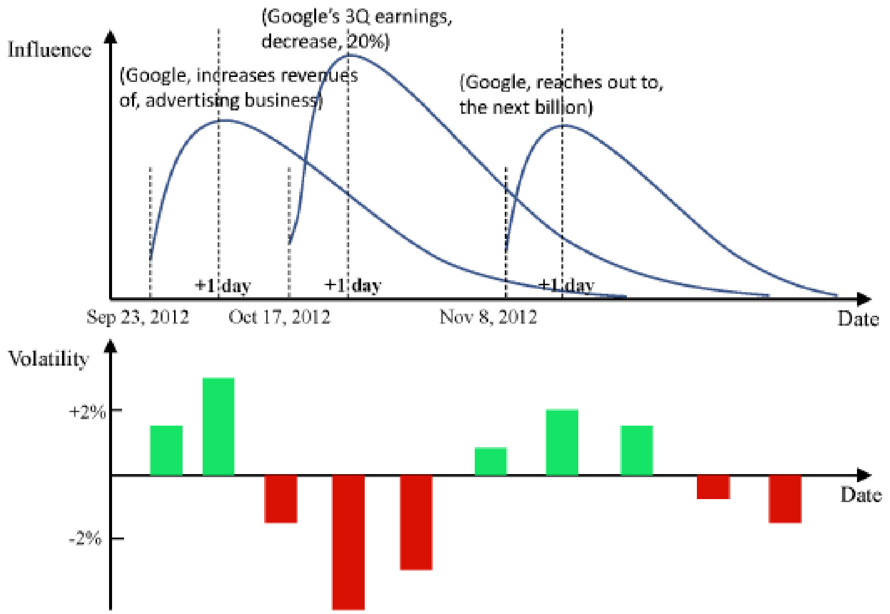
Deep Learning for Event-Driven Stock Prediction

Xiao Ding[†*], Yue Zhang[‡], Ting Liu[†], Junwen Duan[†]
[†]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{xding, tliu, jwduan}@ir.hit.edu.cn
[‡]Singapore University of Technology and Design
yue_zhang@sutd.edu.sg

Summary

- What
  - Use news events and deep learning to predict stock trend
- Why
  - Structured events are sparse
  - Diminishing effects of events
- How
  - Make event representations more dense
    - Neural tensor network
  - Combine short-term and long-term effects
    - Deep convolutional neural network

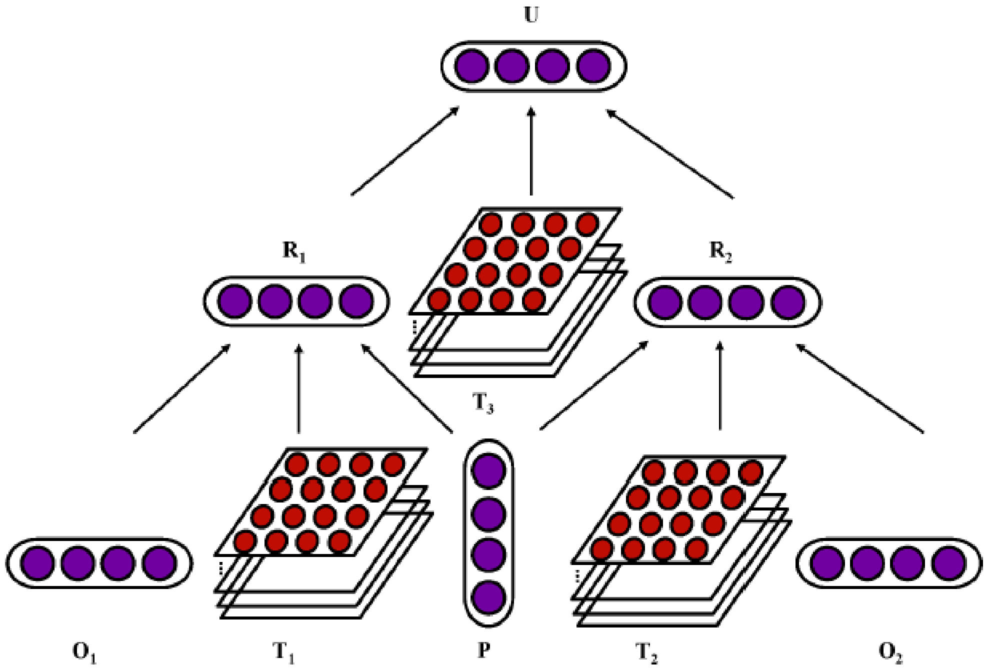
(Actor = Nvidia fourth quarter results, Action = miss, Object = views)
(Actor = Delta profit, Action = didn't reach, Object = estimates)



Method

$$R_1 = f(O_1^T T_1^{[1:k]} P + W \begin{bmatrix} O_1 \\ P \end{bmatrix} + b)$$

- Event representation and extraction
- Event embedding



Algorithm 1: Event Embedding Training Process

Input: $\mathcal{E} = (E_1, E_2, \cdots, E_n)$ a set of event tuples; the model $EELM$

Output: updated model $EELM'$

random replace the event argument and got the corrupted event tuple

$\mathcal{E}^r \leftarrow (E_1^r, E_2^r, \cdots, E_n^r)$

while $\mathcal{E} \neq [\,]$ **do**

  $loss \leftarrow max(0, 1 - f(E_i) + f(E_i^r) + \lambda \|\mathbf{\Phi}\|_2^2$

  if $loss > 0$ **then**

    $Update(\mathbf{\Phi})$

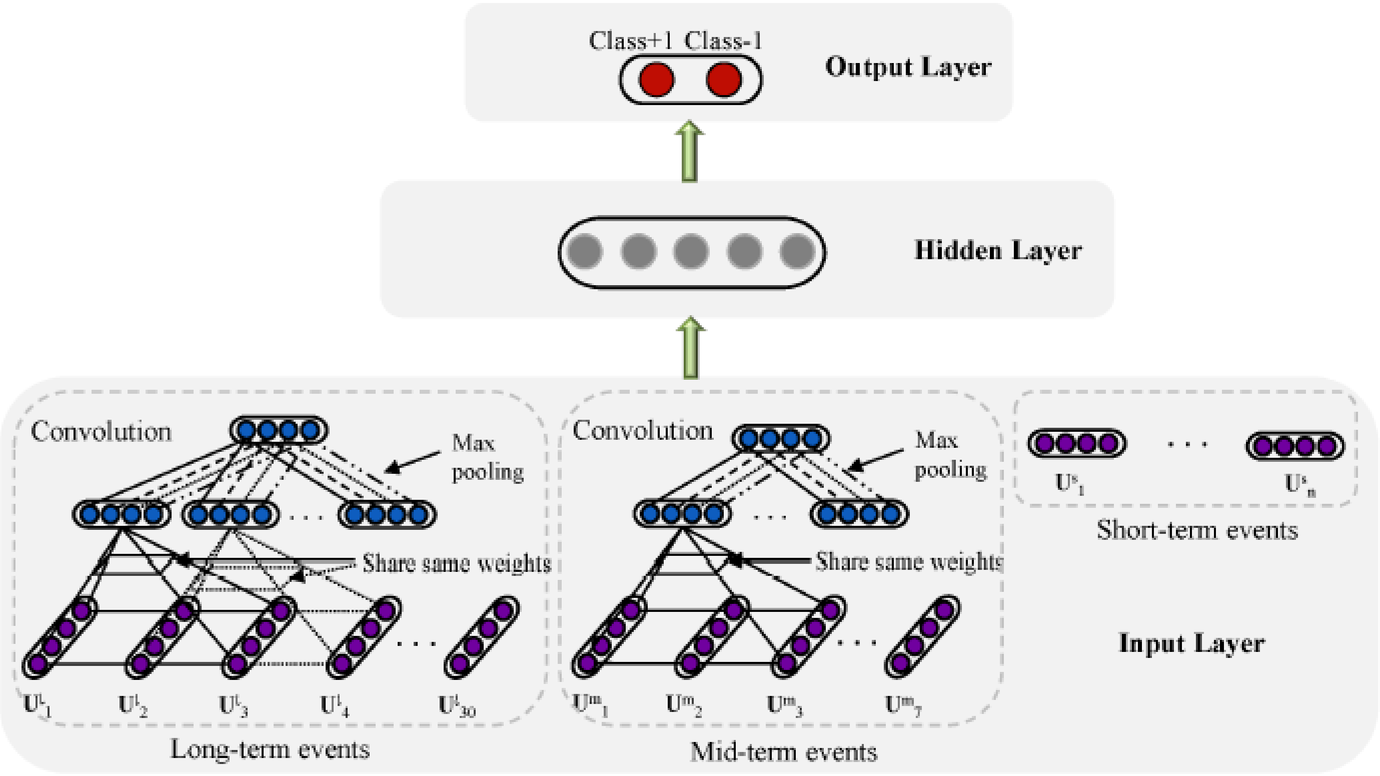
  else

    $\mathcal{E} \leftarrow \mathcal{E}/\{E_i\}$

return $EELM$

Method (Cont.)

- Deep Prediction Model



$$Q_j = W_1^T U_{j-l+1:j}$$

$$V_j = \max Q(j, \cdot),$$

$$V^C = (V^l, V^m, V^s)$$

Experiment

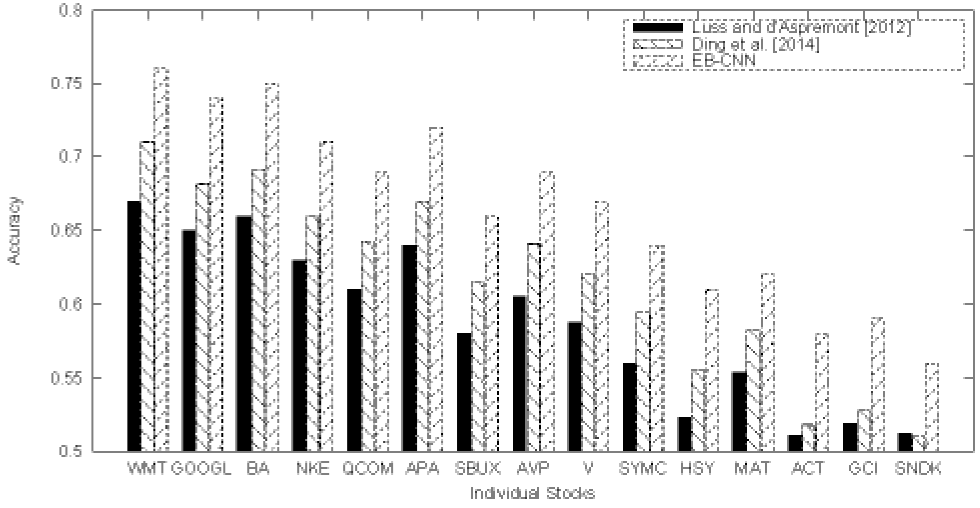
- Data and Metrics
- Comparisons

- **WB-NN**: word embeddings input and standard neural network prediction model [Ding *et al.*, 2014];
- **WB-CNN**: word embeddings input and convolutional neural network prediction model (this paper);
- **E-CNN**: structured events tuple [Ding *et al.*, 2014] input and convolutional neural network prediction model (this paper);
- **EB-NN**: event embeddings input (this paper) and standard neural network prediction model [Ding *et al.*, 2014];
- **EB-CNN**: event embeddings input and convolutional neural network prediction model (this paper).

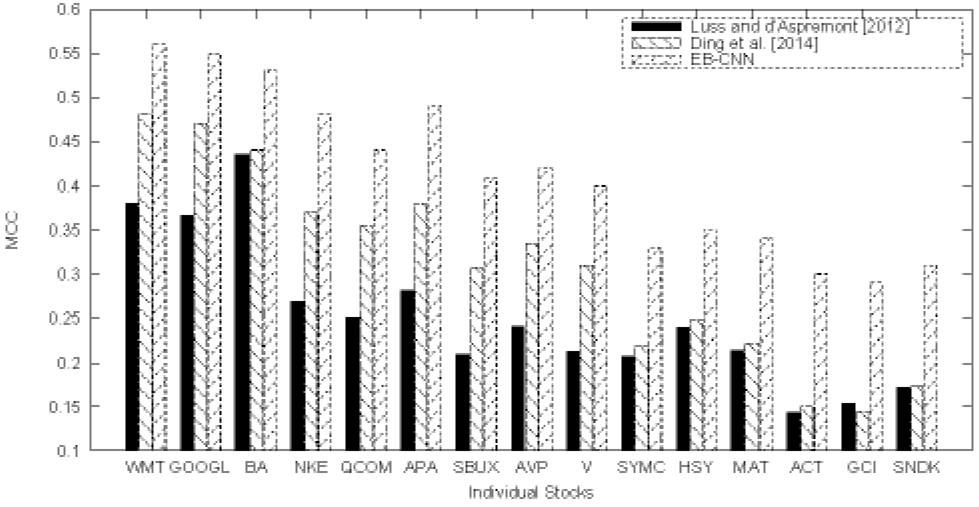
Experiment (Cont.)

	Acc	MCC
Luss and d'Aspremont [2012]	56.42%	0.0711
Ding et al. [2014] (E-NN)	58.94%	0.1649
WB-NN	60.25%	0.1958
WB-CNN	61.73%	0.2147
E-CNN	61.45%	0.2036
EB-NN	62.84%	0.3472
EB-CNN	**65.08%**	**0.4357**

Experiment (Cont.)



(a) Accuarcy



(b) MCC

Experiment (Cont.)

	Index Prediction		Individual Stock Prediction		
	Acc	MCC	Acc	MCC	Profit
Luss [2012]	56.38%	0.07	58.74%	0.25	$8,671
Ding [2014]	58.83%	0.16	61.47%	0.31	$10,375
EB-CNN	64.21%	0.40	65.48%	0.41	$16,774