1.论文信息

论文名：Deep Learning for Event-Driven Stock Prediction

2.概述

股价会受新闻或事件影响，最近的论文开始使用 NLP 技术对金融新闻进行处理以预测股价波动了，但只使用新闻文本的简单特征无法获取结构性联系。比如“Microsoft sues Barnes & Noble.”如果只使用 term-level 特征{“Microsoft”, “sues”, “Barnes”, “Noble”}，则难以准确的预测这两家公司的股价变动，因为无法分辨原告和被告。而使用 open information extraction (Open IE)，能更好的获取事件的主体和客体。

使用事件的结构表征（structured representations）的一个缺点是稀疏度增大，这可能会限制预测能力，我们通过使用事件嵌入（event embeddings）来解决这个问题（事件嵌入是将事件的稀疏向量表示转换为密集、连续的向量空间，使能够识别相似性）。通过训练使得相似的事件对应的向量相似，即使它们的单词完全不同。比如使 (Actor = Nvidia fourth quarter results, Action= miss, Object = views) and (Actor = Delta profit, Action =didn’treach, Object=estimates)对于的向量相似，都表达了公司业绩未达到预期的消极意义。

本文使用 novel neural tensor network (NTN)来训练 event embeddings。对于预测模型，则提出使用 deep CNN 获取新闻事件在一段时间内的影响。研究发现时事对股市波动的影响具有递减效应。按天预测效果比按周预测和按月预测要好，但即使长期事件的影响会随时间减弱，它仍会影响股市。本文将历史新闻事件看做按天划分的事件序列，使用 CNN 和池化层来提取最有代表性的全局特征，然后使用正反馈神经网络，通过一个隐藏层和一个输出层将全局特征和股市变动趋势联系起来。

实验结果表明 event embeddings 能有效解决大规模金融新闻数据集的事件稀疏问题；CNN 模型能通过利用长期事件提高预测效果。

3.模型

1）Event Representation and Extraction

把一个 event 表示成一个元组 E = (O1,P,O2,T)，
O1 是主体，P 是谓语，O2 是客体，T 是时间
比如“Jan 13, 2014 - Google Acquires Smart Thermostat Maker Nest For for $3.2 billion.”可以表示为 E= (Actor = Google, Action = acquires, Object =Nest, Time = Jan 13, 2014).

使用 Open IE 技术和依存句法分析来从文本中提取结构化事件：给定新闻文本中的一个句子，先用 ReVerb 方法提取该事件的候选元组(O1’,P’,O2’)，然后用 ZPar 对这个句子进行语法分析以提取主体、客体和谓语，如果与(O1’,P’,O2’)不相符，则舍弃这个元组。由于新闻句子多且冗余，因此这个方法最终肯定能够获取新闻的主要事件元组。

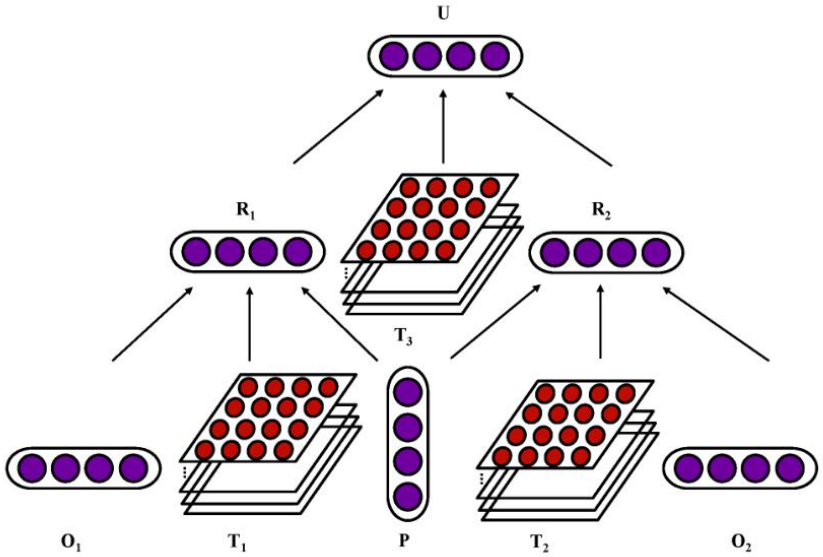
2）Event Embedding

此任务与之前的从知识库的多元关系数据中学习分布式表示任务相关联，即学习嵌入(e1,R,e2)，其中 e1、e2 是实体而 R 是二者的关系类型。但 structured event embedding 的学习与之有两处

明显的不同：

首先，知识库中的关系类型数量有限，因此使用矩阵或张量来代表关系类型即可，为每种关系类型训练出一个矩阵或张量。但在此次任务中，我们使用 Open IE 技术提取事件，事件类型是未知的，因此难以为每种事件类型训练一个特定的模型。为此我们将 P 表示成一个向量，其与事件参数维度相同。

其次， (e1,R,e2)中 R 是对称的，e1、e2 可互换位置，但 structured event embedding 中 O1、O2 不可互换。因此我们使用 novel neural tensor network，分别使用张量 T1、T2 学习 O1、O2 的角色。 O1T1P 和 PT2O2 分别被用于构建 role-dependent embeddings R1 和 R2。 第三个张量 T3 则用于 R1 与 R2 的语义合成，最终得到事件元组 E = (O1,P,O2)对应的 structured embedding U 。



3）Neural Tensor Network

neural tensor network 的输入是 word embeddings，输出是 event embeddings。
先使用 skip-gram 算法从大规模金融新闻语料库中学习到 d 维（d=100）word representation。
由于一般参数包含多个单词，我们将 event 的 actor、action 和 object 表示为对应 word embeddings 的平均值

From Figure 2, $R_1 \in \mathbb{R}^d$ is computed by:

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

其中 $T_1^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ 是一个张量

双线性张量积 $O_1^T T_1^{[1:k]} P$ 是一个向量 $r \in \mathbb{R}^k$，其中 $r_i = O_1^T T_1^{[i]} P$，i=1,...,k

其他参数是标准的正反馈神经网络，$W \in \mathbb{R}^{k \times 2d}$ 是权值矩阵，$b \in \mathbb{R}^k$ 是偏差向量，

$f = tanh$ 是激活函数。
R2 和 U 的计算方法同理。
我们也试过将随机初始化的 word 向量作为 NTN 的输入。但使用预先训练过的 word embeddings 作为输入效果更好。

4）Training

训练集含一千万个事件样本，对每个训练样本进行 N 次迭代，其中一个样本就是一个事件元组 E=(O1,p,O2)，其提取方法如上文所述。
每次迭代过程如下：

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 $EELM$

每次迭代随机替换事件参数，获取 corrupted event tuple
得到更新后的事件元组集合
遍历集合中所有事件元组，计算 loss，loss>0 时更新Φ。

训练集中事件元组的权重应该比 corrupted 元组高，corrupted tuple 指其中一个事件参数被一个随机参数替换，即对元组 E=(O1,p,O2)中的所有 O1 的 word，随机选取训练集中的 word 替代，得到 Er=(O1r,P,O2)。
损失函数定义为：

$$loss(E, E^r) = \max(0, 1 - f(E) + f(E^r)) + \lambda \| \Phi \|_2^2,$$

参数集合Φ =(T1 ,T2 ,T3 ,W,b)，标准 L2 正规化权重 λ=0.0001
如果 $loss(E, E^r) = \max(0, 1 - f(E) + f(E^r))$ 值为 0，则继续处理下一个事件元组。否则使用 BP（反向传播）算法更新参数以最小化损失。设迭代次数 N=500

5）Deep Prediction Model

把过去一个月内的 event 看作 long-term-event，把过去一周内的 event 看作 mid-term event，把过去一天内的 event 看作 short-term event。然后用 CNN 学习这三种 event 对股市的影响。

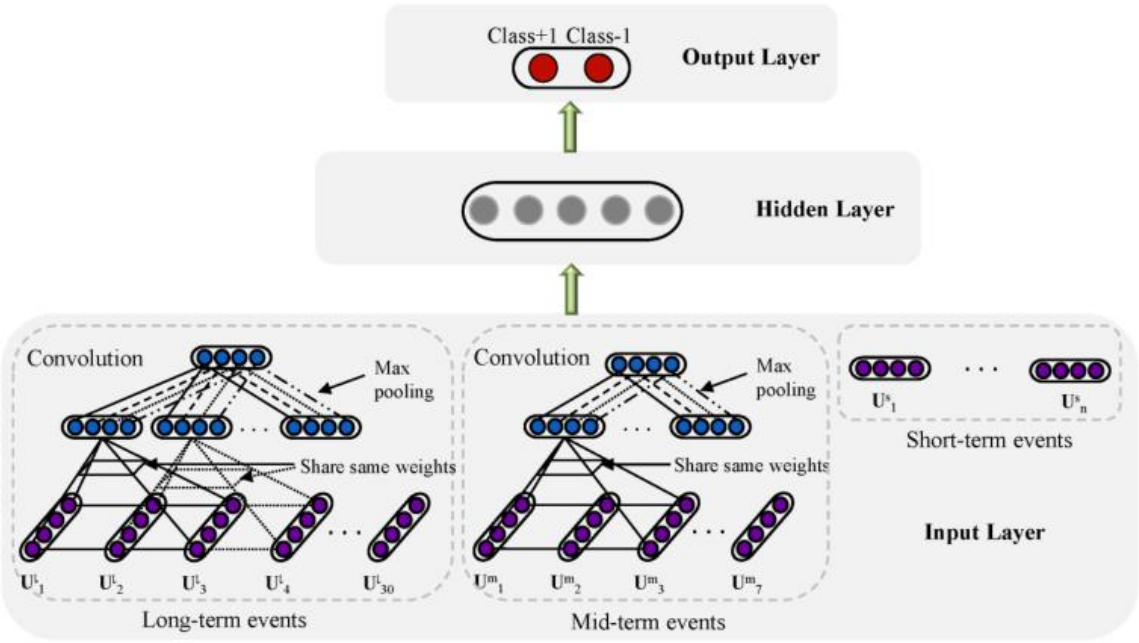


Figure 3: Architecture of the prediction model based on a deep convolutional neural network.

将按时间排序的 event embeddings 序列作为模型输入，每一天的 event embeddings 取平均数作为输入单元 U，模型输出是二元分类，Class +1 代表股价上涨，Class -1 代表股价下跌。
对于 long-term news 和 mid-term news，使用窄卷积将 l（l=3）个相邻 event 组合起来。相当于使用滑动窗口的特征提取操作，它可以通过将一个窗口内的向量结合起来获取局部信息。
预测任务需要利用所有局部特征来全局性的预测股价变动。因此在卷积层上方使用最大池化层，使网络仅保留最有用的局部特征。只对 long-term and mid-term event embeddings 进行卷积。

给定输入的 event embeddings 列表 U =(U1 ,U2 , ••• ,Un)， $U_i \in \hat{\mathbb{R}}^d$

一维卷积函数取权值向量 $W_1 \in \mathbb{R}^l$ 与 U 中每个滑动窗口内的 l 个 event embeddings 的点积以获取一个新的序列 Q：

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

为了判定出全局最优代表性的特征，我们对 Q 进行最大池化操作：

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

$Q(j, \cdot)$ 代表矩阵 Q 的第 j 行。最大池化后我们得到特征向量 V。依此操作，分别获得 long-term and mid-term events 对应的特征向量 Vl 和 Vm。对于 short-term evemts 则直接使用 averaged event embeddings Us 获得特征向量 Vs。特征层将以上三者结合起来得到特征向量 Vc=(Vl,Vm,Vs)。
为了将特征向量 Vc 与股价联系起来，本文使用了单隐藏层和单输出层的正反馈神经网络。
设输出层的输入为 netcls，输出为 ycls (cls∈{+1,−1})，设隐藏层的神经元向量为 Y，

则 $y_{cls} = f(net_{cls}) = \sigma(W_3^T \cdot Y)$，其中 $Y = \sigma(W_2^T \cdot V^C)$

其中 σ 是 sigmoid 函数，W2 是隐藏层与特征层之间的权值向量，W3 是输出层神经元 cls 与隐藏层之间的权值向量。

4.实验

1）数据集&评估指标

数据集： Reuters、Bloomberg 上的金融新闻，日期为 2006.10~2013.11。
研究发现新闻标题用于预测比新闻内容效果更好，本实验只从新闻标题中提取事件，预测 S&P 500 index 和个人股。从雅虎金融获取目录和价格。
训练集、验证集、测试集分割如下：

	Training	Development	Test
#documents	442,933	110,733	110,733
#words	333,287,477	83,247,132	83,321,869
#events	295,791	34,868	35,603
time interval	02/10/2006 - 18/06/2012	19/06/2012 - 21/02/2013	22/02/2013 - 21/11/2013

Table 1: Statistics of datasets.

评估指标：Acc、MCC（Matthews Correlation Cofficient）、利润率

2）设置对比实验

两个 Baselines 算法 ：
（1）使用词袋代表新闻文档，使用 SVMs 构建预测模型；
（2）使用 event tuple E=(O1,p,O2)代表新闻文档，使用标准正反馈神经网络来探究 events 与股价变动之间的隐藏关系。
本文使用张量神经网络从新闻文本中学习 event embeddings，然后使用深度 CNN 构建预测模型。
为了做更详细的对比分析，构建了以下五个模型：
WB-NN: word embeddings input and 标准 NN prediction model ;
WB-CNN: word embeddings input and CNN prediction model;
E-CNN: structured events tuple input and CNN prediction model;
EB-NN: event embeddings input and 标准 NN prediction model;
EB-CNN: event embeddings input and CNN prediction model
下划线部分是本文所用算法。
word embedding input (WB)包括文档所有单词之和，解决了 word-based 输入的稀疏问题，因此可以作为一个 baseline embedding 方法。标准 NN 算法用于与 deep CNN 做对比。

3）验证集结果

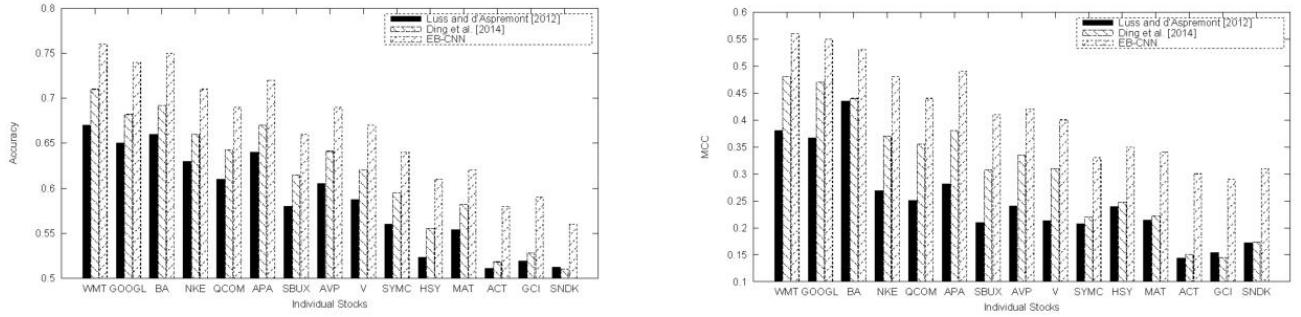
S&P 500 Index Prediction（股市整体）

	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**

Table 2: Development results of index prediction.

1）word-embedding **vs** event-embedding，event-embedding 效果更好（e.g. WB-NN vs EB-NN and WB-CNN vs EB-CNN）
原因：event 比 word 更适合作为特征进行股价变动预测
2）event-embedding **vs** structed event tuple，event-embedding 效果更好（e.g. E-CNN vs EB-CNN，E-NN vs EB-NN）
原因：
a）低维稠密向量有效地缓解特征稀疏问题，word embeddings 效果也比 structed event tuple 好（WB-NN vs E-NN），因为 word embeddings 也是使用低维稠密向量；这说明了降低稀疏度的重要意义，甚至比提取结构化信息（event）更重要。
b）通过进行 word embeddings 的语义合成，可以从 event embeddings 间学习到更深层的语义联系。
3）CNN 比 NN 效果好 (e.g. WB-CNN vs WB-NN,EB-CNN vs EB-NN, and E-CNN vs E-NN)
原因：CNN 可以分析长期历史 events 的影响、提取最优代表性的特征向量。

Individual Stock Prediction（个别公司股价）
在验证集上与 baseline 算法作 Individual Stock Prediction 的对比实验。选取 15 家公司（排名高、中、低都有）做预测，实验结果为：



(a) Accuarcy (b) MCC

Figure 4: Development results of individual stock prediction (companies are named by their ticker symbols).

分析：(1)无论是 S&P 500 index prediction 还是 individual stock prediction，本文模型（EB-CNN）表现比两种 baseline 方法更好。
（2）本文模型在财富排名较低的公司股价预测上所取得的效果提升更显著，这些公司相关的新闻更少。两种 baseline 算法在预测 low-ranking 公司股价时效果大幅降低，而我们的算法由于考虑到了 weekly news 和 monthly news 影响的递减效应，即使缺少 daily news，仍能取得相

对更准确的预测结果。

4）利润

模拟股市交易，如果模型预测某支股票会在第二天上涨，就以开盘价买下它的价值 $10000 的股票，然后持有这支股票一天的时间，如果在这一天里发现能赚取 2%或更多的利润，就马上抛售，否则在这一天的最后以收盘价出售；当模型预测某支股票会下跌时，就借股票出售，在这一天内如果交易者能够以比卖空价格低 1%的价格买入股票，就立即买入股票归还，否则最后以收盘价买入股票归还。（就是卖空赚差价，先高价卖掉股票，然后又低价买回相同份额的股票）。
以下是在验证集上得到的平均累计利润

	Average Profit
Luss and d'Aspremont [2012]	$8,694
Ding et al. [2014]	$10,456
EB-CNN	$16,785

Table 3: Averaged profit of 15 individual companies.

可见本文算法赚取的利润更高，对于两个 baseline 算法，发现当公司当天没有新闻时，就无法进行预测，因为他们的模型无法利用 long-term 和 mid-term 新闻，这不会降低 Acc 和 MCC，但会降低实际利润。
为了在统计意义上证实模型成果，本文进行了随机测试，即随机购买或出售 1000 次，结果平均损失了$9,865。

5）测试集结果

以下在测试集上的实验结果，可见本文模型的优越性和鲁棒性

	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

Table 4: Final results on the test dataset.

与 Lavrenko 的算法的对比：

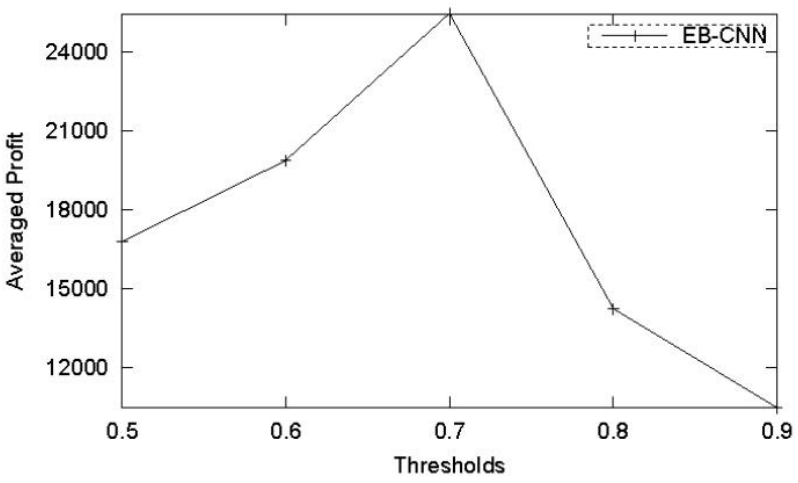
Stock	Profit of Lavrenko et al. [2000]	Profit of EBCNN
IBM	$47,000	$42,000
Lucent	$20,000	$27,000
Yahoo	$19,000	$32,000
Amazon	$14,000	$35,000
Disney	-$53,000	$7,000
AOL	-$18,000	$14,000
Intel	-$14,000	$8,000
Oracle	-$13,000	$17,000

Table 5: Profit compared with Lavrenko et al. [2000]. (there are 4 negative profit stocks out of 15 which are not included in this table)

除 IBM 外本文算法都得到更高利润。

6）交易策略

根据分类概率购买或出售股票，如果预测上涨的概率超过阈值β就买入$10,000 的股票，预测下跌的概率超过阈值β就卖空$10,000 的股票。不然不买不卖，结果如下：



可见阈值为 0.7 时所获利润最高。这说明了交易策略也会对利润率有较大影响，这是一个改进方向。

5.评价

1）优点

a）通过使用张量神经网络，能学习到对于谓语来说，哪个是主体，哪个是客体。
b）使用嵌入方法 embedding 往往能提高效率，因为低维稠密向量更高效、是更具代表性的特征。
c）鲁棒性好，由于使用 deep CNN，模型能利用 long-term 和 mid-term 事件的递减影响，即使当天没有相关新闻也能做出合理的预测，这有利于对小公司股价预测，其他算法则是只能利用当天新闻进行预测，因此如果当天没有新闻则无法预测。

2）缺点

a）本文算法那能学习到主体、谓语、客体信息，但无法学习其他信息比如状语。比如一般新闻事件的可以分为正面、负面两种，但各自也有不同的程度，即使是负面信息也有可能只是不痛不痒，如果历史事件都是正面信息的话，应该继续买进股票。

3）改进方向

a）事件元组设为 E=(O1,P,O2,A)，其中 A 代表状语，在 NTN 中适当调整增加张量学习状语信息。
b）不仅仅根据新闻事件进行预测，还可以参考时间序列预测方法，根据过去一段时间内的股价变动数据，预测股价走势，用作辅助。
c）研究更多不同的交易策略，比如可以考虑长期持股，而不是必须当天买当天卖，使能够更有效的配合本文算法，取得最大化的效益。