Using Structured Events to Predict Stock Price Movement

Ekaterina Kochmar

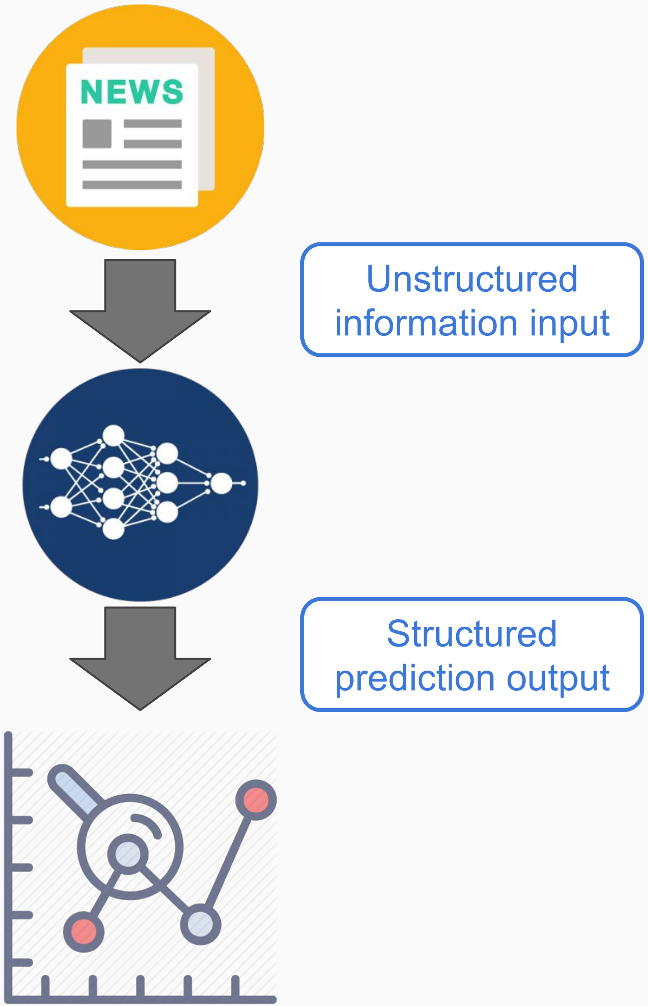
A paper by: Ding et al. (2014). *Using Structured Events to Predict Stock Price Movement: An Empirical Investigation*

URL: https://www.emnlp2014.org/papers/pdf/EMNLP2014148.pdf

“News events influence the trends of stock price movements”

Today we are going to discuss:

- How to extract the structured information from unstructured data
- How to use this information in a Machine Learning framework to make optimal predictions



Background

A bit of history

- **Idea**: a lot of relevant information comes in the form of natural language text, e.g. news. Events reported in financial news are important for stock price movement prediction
- Prediction is valuable to investors, public companies, governments
- *Random Walk Theory* (1973): prices are determined randomly ➔ impossible to outperform the market
- *Efficient Market Hypothesis* (1965): the price of a security reflects all of the information available and everyone has a certain access to this information

A bit of history

- Early studies used **bag-of-words** approach – doesn't help to define the relations between entities
- Later studies that focused on events struggled with **scalability**
- Emotions and sentiment matter: negative words carry the signal about the future stock market moves, however this is **subjective**

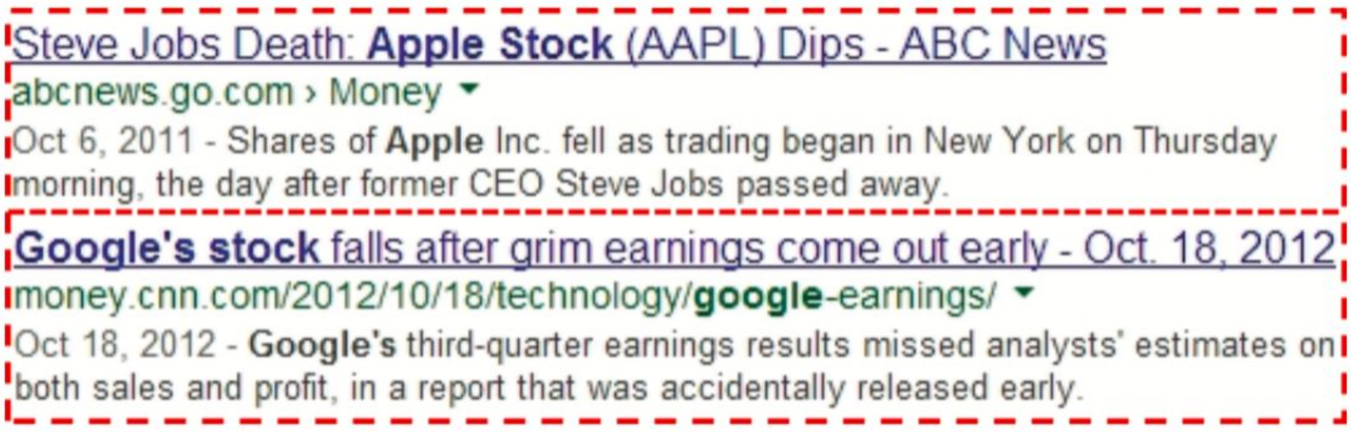
The approach taken in this paper is objective, event-based and does not suffer from scalability problems

Why Natural Language?

Natural Language

- Speaking
- Listening
- Writing
- Reading
- Planning
- Dreaming
- Discussing
- Conveying information
- etc.

Natural Language in stock market prediction

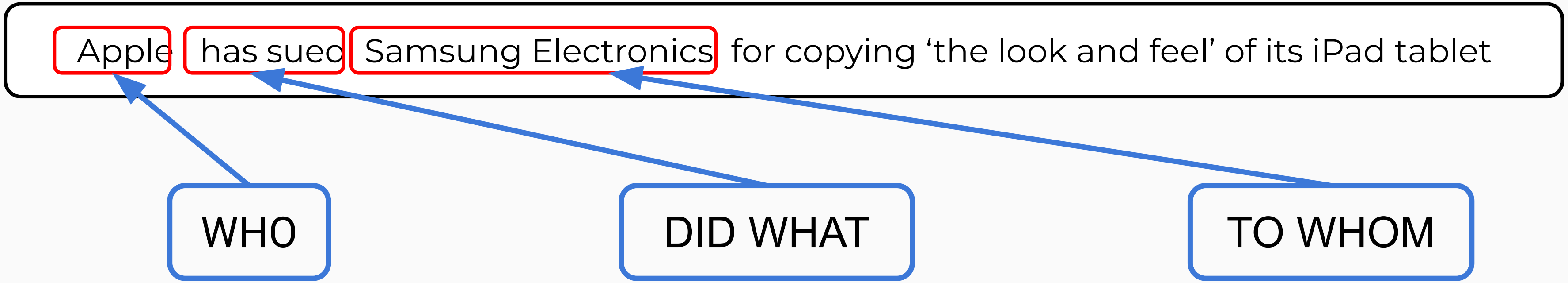


- Shares of Apple Inc. fell after news piece about the death of Apple's former CEO
- Google's stock fell after grim earnings came out

Challenges for Natural Language Processing (NLP)

- This information is unstructured – how can we make sense of it?
- Three approaches attempted in the past:
    - *Bags-of-words*: {Apple, has, sued, Samsung, Electronics, for, copying}
    - *Noun phrases*: {Apple, Samsung Electronics, copying}
    - *Named entities*: {Apple, Samsung Electronics}
- Alternative attempted in this work – **events model**

Events model

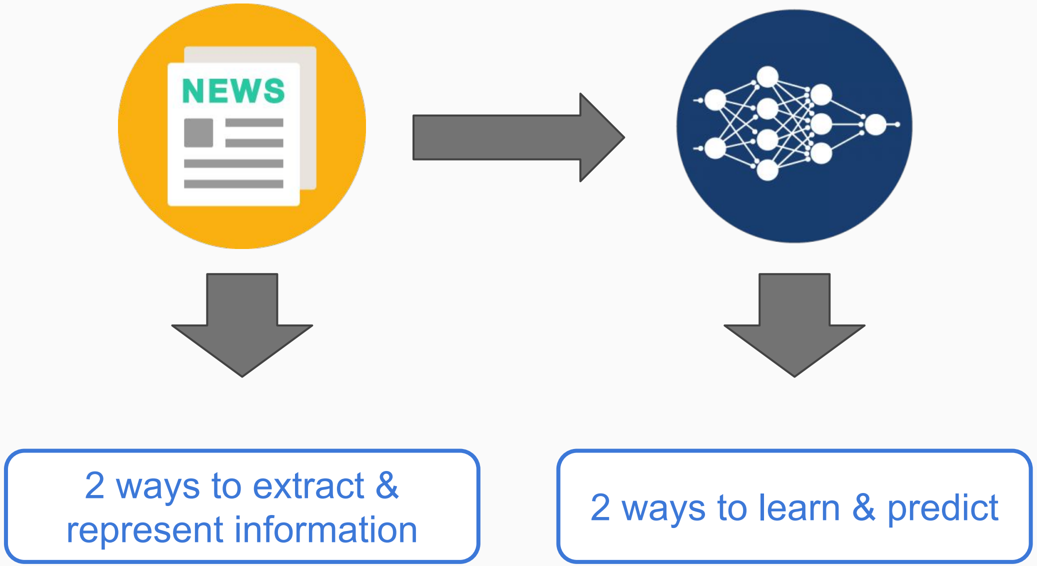


- The "Who" bit is called **actor** O_1
- The "did what" bit is called **relation** or **predicate** P
- The "to whom" bit is called **object** O_2

Method

Method

- NLP bit: information extraction & representation
- ML bit: prediction



NLP (1): Event representation

Sep 3, 2013 - Microsoft agrees to buy Nokia's mobile phone business for $7.2 billion.

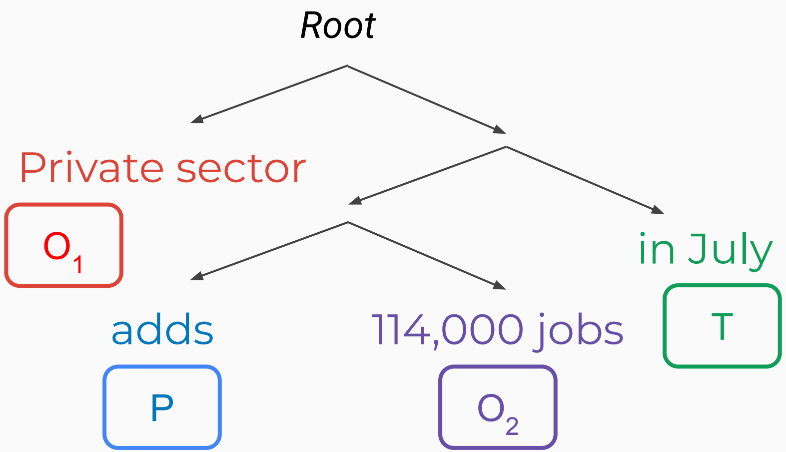
- Build an event model $E = (O_1, P, O_2, T)$
  - O_1 = *Microsoft*
  - P = *buy*
  - O_2 = *Nokia's mobile phone business*
  - T = *Sep 3, 2013*

NLP (2): Event extraction

Instant view: Private sector adds 114,000 jobs in July: ADP

- How to extract structured information from unstructured input?
- **Bag-of-words**: simply list all words {Instant, view, Private, …}
- Predefined event type (template) – doesn’t generalise
- Alternative – Open IE (Banko et al., 2007; etc.) framework

NLP (2): Event extraction



- Apply NLP tools – parsing: identify the relations between words
  - *P* has to denote an action (verb)
  - Both O_1 and O_2 have to denote some objects (nouns)

NLP (3): Event generalisation

Microsoft swallows Nokia’s phone business for $7.2 billion

=

Microsoft purchases Nokia’s phone business for $7.2 billion

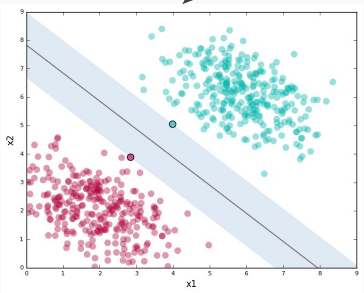
- How can we establish equivalence between different forms?
    - *WordNet*: an hierarchical database for all words
    - *FrameNet*: classes for verbs. E.g., *add = multiply_class*

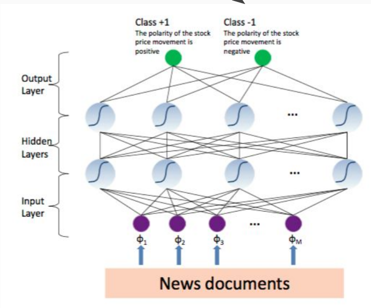
ML (1): Overview

Extract features

Learn function

Bag-of-words vs event-based





ML (1): Bag-of-words feature representations

- **Bag-of-words** features: offset by Tf-Idf
  - Offset by term-frequency (TF): TF = freq(t) / length(d)
  - Offset by inverse-document-frequency (IDF): log(N / documents with t)
- Example: we see "Microsoft" 2 times in document d_1 and 2 times in document d_2
  - If feature f_1="Microsoft", should we include [2, …] in the feature vector of d_1 and d_2?
  - Suppose length(d_1)=100 words and length(d_1)=200 words – is there a difference in contribution of f_1="Microsoft" to d_1 and d_2?
  - Suppose we have 100 documents in the whole dataset and they all mention "Microsoft" – how informative is this word as a feature then?

ML (1): Bag-of-words feature representations

- **TF:** Offset by term frequency: TF = freq(t) / length(d)
    - Contribution of f_1="Microsoft" to d_1 is equal to $tf(f_1,d_1) = 2/100 = 0.02$
    - Contribution of f_1="Microsoft" to d_2 is equal to $tf(f_1,d_2) = 2/200 = 0.01$
    - The longer the document, the more word occurrences we'll see!
- **IDF**: Offset by inverse document frequency log(N / documents with t)
    - If each document in the collection has feature f_1="Microsoft" present, its contribution is not very high: $idf(f_1) = \log(100 / 100) = 0$
- The final weight of the feature in each feature vector is defined not by the absolute occurrence count, but by tf * idf

ML (1): Event-based feature representations

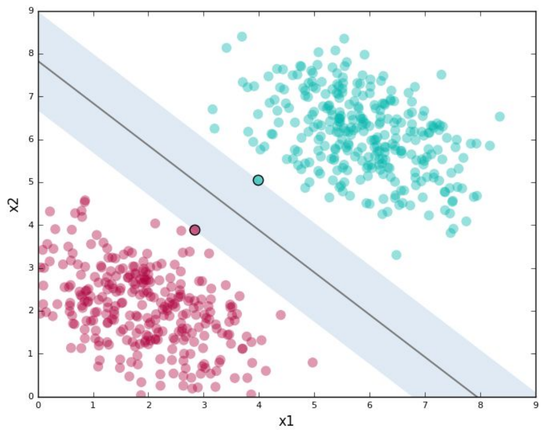
- **Events-based** features: + sparseness reduction applied via FrameNet
  - O_1 = “*Microsoft*”
  - P = “*buys*”
  - O_2 = “*Nokia’s business*”
  - O_1 + P = “*Microsoft buys*”
  - P + O_2 = “*buys Nokia’s business*”
  - O_1 + P + O_2 = “*Microsoft buys Nokia’s business*”

ML (1): Event-based feature representations

- For example, f_1=("*Microsoft*", "*buys*", "*Nokia's business*"), ... , f_{100}=("*Microsoft*", "*buys*"), ..., f_{400}=("*Microsoft*"), and so on
- Note that f_i=("*Microsoft*") as O_1 and f_j=("*Microsoft*") as O_2 will be different features
- For each text, the feature vector will register which of the events are present: e.g., if f_1=("*Microsoft*", "*buys*", "*Nokia's business*") and the tuple is present in document d_1, then feature vector will be [1, ...], and [0, ...] otherwise

ML (2): Linear model – Support Vector Machines

Class = -1
(all documents that predict decrease)



Class = +1
(all documents that predict increase in price)

- Training set: $(d_1, y_1), (d_2, y_2), \ldots, (d_N, y_N)$
- Learn: $w * \boldsymbol{\Phi}\,(d_n, y_n)$
- Predict: $y = argmax\ \{Class = -1,\ Class = +1\}$
- Using the labelled training data, learn weights in order to build the separation boundary

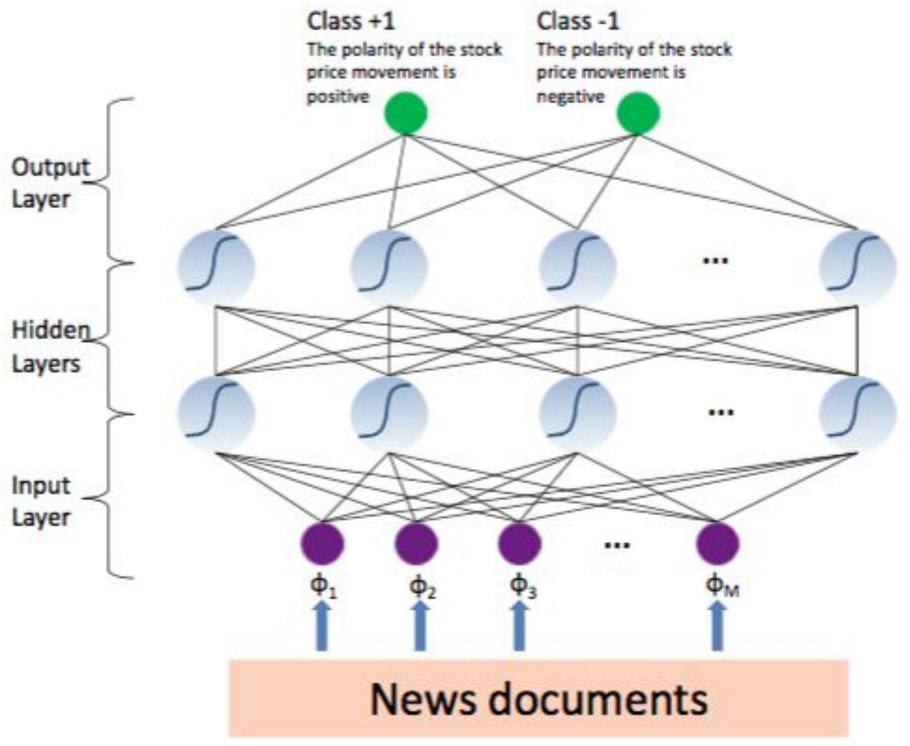
ML (3): Nonlinear model – Neural Network

$$y_{cls} \ (cls \in \{+1, -1\})$$

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

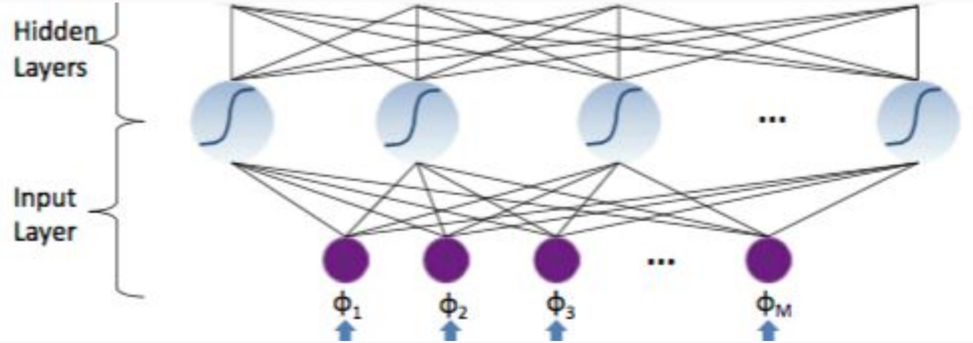
$$\boxed{y_{2k}} = \sigma(\mathbf{w_{2k}} \cdot \mathbf{y_1}) \quad (k \in [1, \ |y_2|])$$

$$\boxed{y_{1j}} = \sigma(\mathbf{w_{1j}} \cdot \mathbf{\Phi}(d_n)) \quad (j \in [1, \ |y_1|])$$



ML (3): Nonlinear model – Neural Network

- **Input:** feature vector **Φ** with values for *M* features in doc
- For the first hidden layer, **learn** matrix (*M x J*) of weights **w1**
- **Output**: first layer of hidden neurons **y1**



$$\Phi = [\Phi_1 , \ldots , \ \Phi_M]$$

$$\Big\downarrow \quad \boxed{\text{“Translate” with w1}}$$

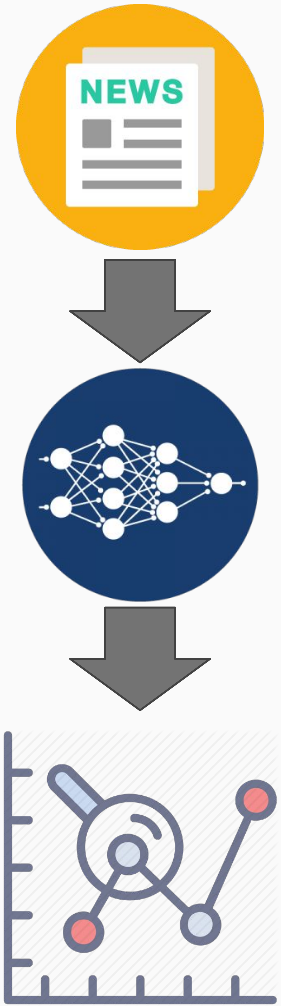
$$y1 = [y1_1 , \ldots , y1_j]$$

$$w1 = \begin{array}{cc} & J \\ \left[\begin{array}{ccc} \text{1st feature to 1st neuron} & & \text{1st feature to j-th neuron} \\ & \ldots & \\ \text{M-th feature to 1st neuron} & & \text{M-th feature to j-th neuron} \end{array}\right] & M \end{array}$$

Experiments

Experiments

- Data
- Evaluation
- Results



Data

- Financial news from Reuters and Bloomberg: titles and contents
- Time period: October 2006 to November 2013
- Data split into train : dev : test = 80% : 10% : 10%

	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

Experimental setup

- 2 x 2 features by methods setup
- x 3 time intervals: short (1 day) / medium (1 week) / long (1 month)

bag-of-words & SVM	event-based & SVM
bag-of-words & Neural Net	event-based & Neural Net

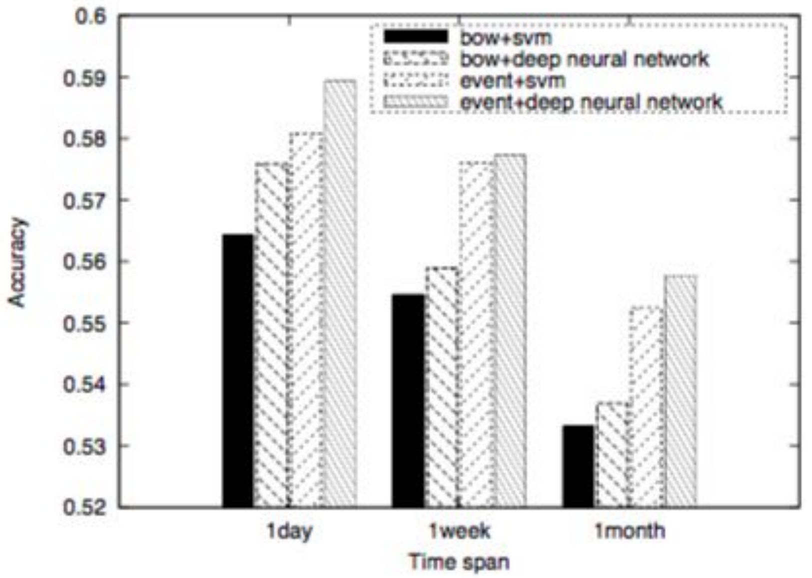
Evaluation strategies

- Accuracy = number of correct predictions / total
- Matthews Correlation Coefficient (MCC):

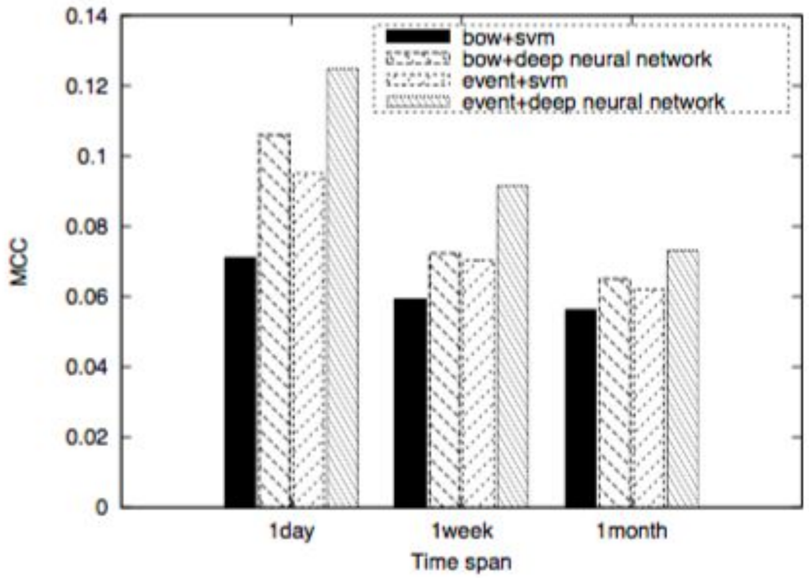
	Predicted Class 1	Predicted Class -1
Actual Class 1	True positives = TP	False negatives = FN
Actual Class -1	False positives = FP	True negatives = TN

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

Results (1): Overall development results



(a) Accuarcy



(b) MCC

Results (2): Take-away messages

1. **Structured (event-based) vs unstructured (bag-of-words)**: structured features consistently outperform; carry essential information
2. **Linear (SVM) vs nonlinear (Neural Net) models**: nonlinear model consistently outperforms; learns hidden relationships
3. **Time interval effects**: short-term volatility easier to predict; many news have immediate effect; historical data is hard to get hold of

Results (3): Neural Network architecture effects

How deep should the model be?

- The deeper the better, but there is a natural constraint on training

		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

Results (4): Amount of data effects

How much data should be used?

- Titles encode most relevant information
- Contents helps less
- There is a huge overlap between the news sources (up to 80%!)

	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

Results (5): Individual stock prediction

Can better prediction be achieved using company / sector / all news?

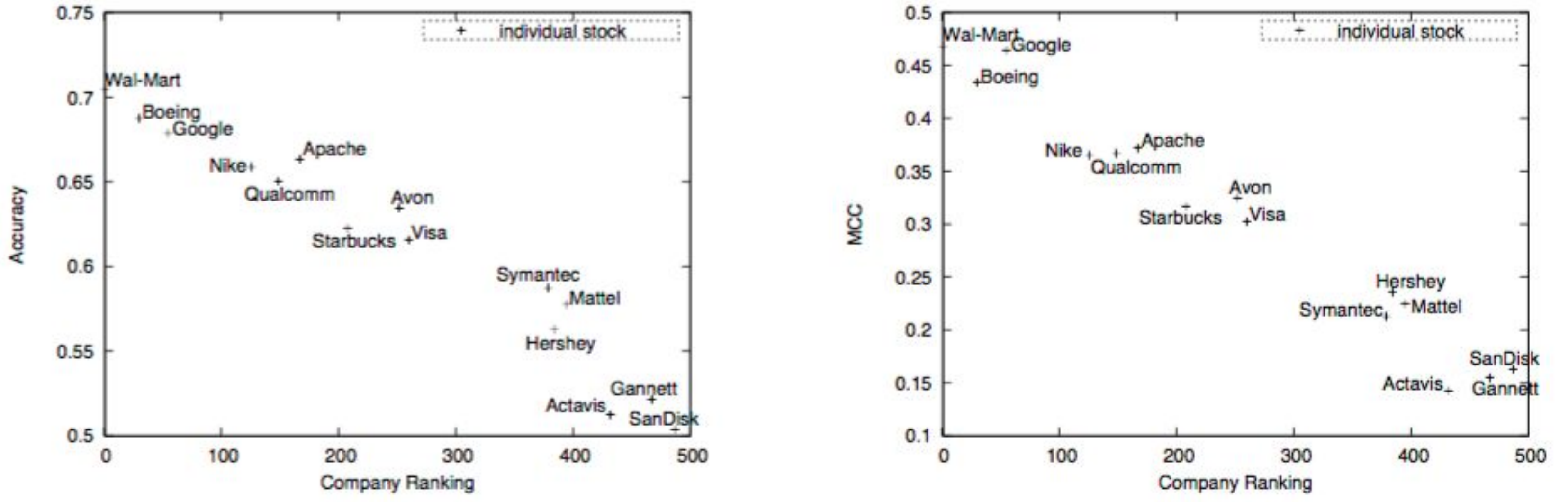
- Company news are very relevant
- Sector and all news damage performance

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

Results (6): Individual stock prediction on 15 companies

Generalisation over 15 companies:

- Amount of available news matters – lower for lower fortune rankings

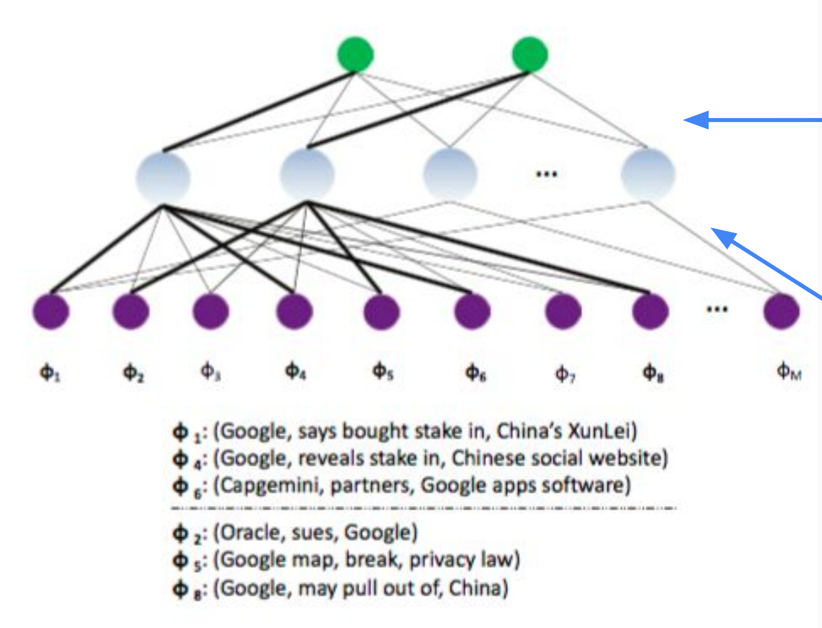


(a) Accuarcy

(b) MCC

Results (7): Towards black box interpretability

- Positive events shown to relate to class +1 prediction
- Negative events shown to relate to class -1 prediction



Here is where hidden units are connected with higher weights to one output class or another

Here is where we can see the relation of the features to the hidden units

Questions?

Contact:

Ekaterina Kochmar

Ekaterina.Kochmar@cl.cam.ac.uk
www.cl.cam.ac.uk/~ek358/