Event extraction & Event embedding

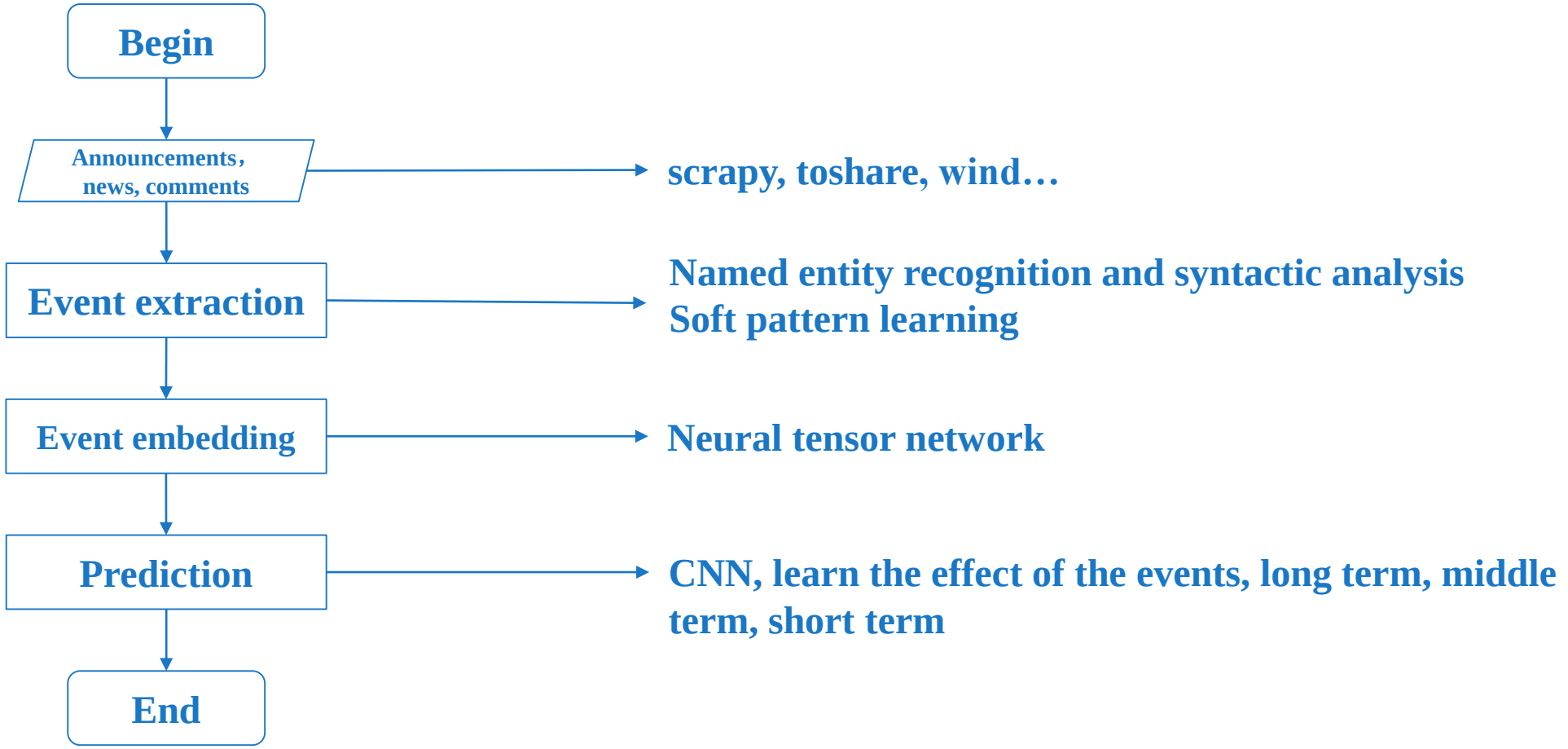
杞坚玮

2017.5.10

What I want to talk about

1. Event representation
2. Event extracting
3. Event embedding
4. Future work
5. Reference

What I want to talk about



1. Event representation

“Microsoft sues Barnes & Noble.”

Traditional models such as bags-of-words:

{“Microsoft”, “sues”, “Barnes”, “Noble”}

Structured model:

{“Actor”: “Microsoft”, “Action”: “sues”, “Object”: “Barnes & Noble”}

1. Event representation

Event: (O_1, P, O_2, T)

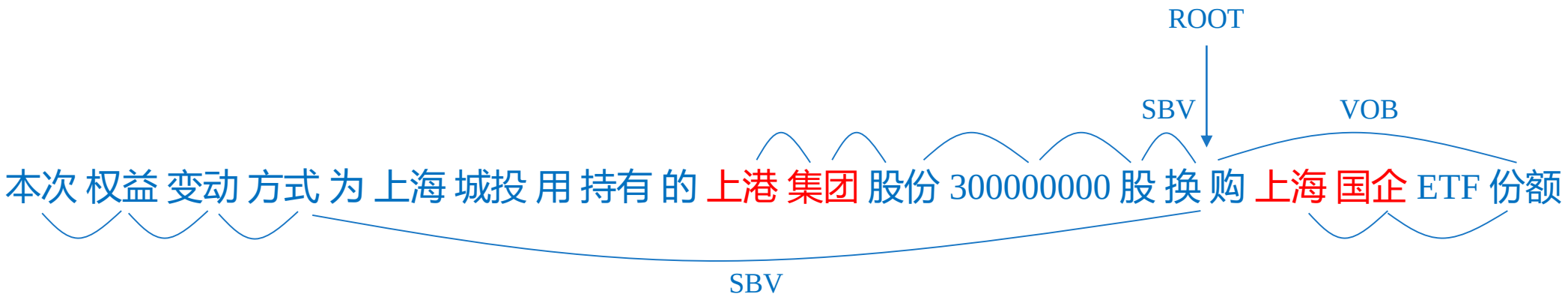
In this tuple, P is the action, O_1 is the actor and O_2 is the object on which the action is performed. T is the timestamp of the event, which is used for aligning stock data with news data.

For example: “Jan 13, 2014 – Google Acquires Smart Thermostat Maker Nest For for $3.2billion”

The event tuple is (Actor = Google, Action = Acquires, Object = Nest, T = Jan 13, 2014)

2. Event extracting

“本次权益变动方式为上海城投用持有的上港集团股份300,000,000股换购上海国企ETF份额”

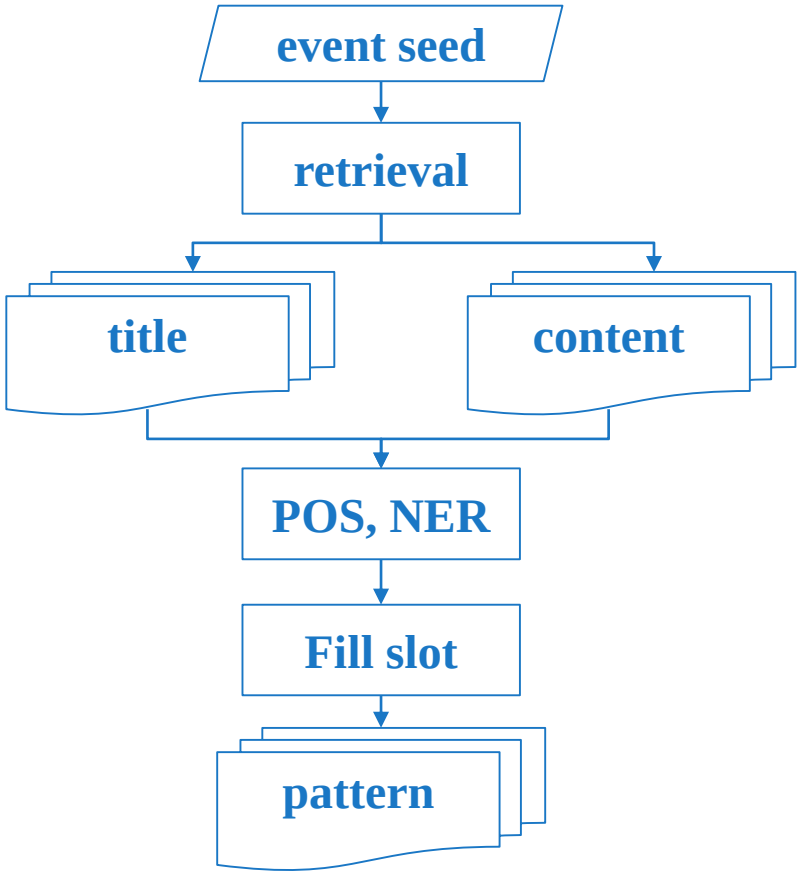


1. Root in interested verb list?
2. Include name entity?
3. Has (x, v, y) pattern?

⇒ Event tuples (‘上港集团’, ‘换购’, ‘上海国企’)

2. Event extracting

Bootstrapping-pattern learning



('上海城投', '换购', '上海国企')

上海城投交易所持有的上海国企份额
上海城投减持所持有的上海国企份额

上海城投/Ni 交易/v 所持有的/d 上海国企/Ni 份额/n
上海城投/Ni 减持/v 所持有的/d 上海国企/Ni 份额/n

[slot1]/Ni 交易/v 所持有的/d [slot2]/Ni 份额/n
[slot1]/Ni 减持/v 所持有的/d [slot2]/Ni 份额/n

2. Event extracting

Bootstrapping-pattern generalization

- Exactly the same, cost is 0
- Entities of the same type, cost is 5
- Non-entity, synonyms, cost is 5, otherwise cost is 10
- None of the above, cost is 10

[slot1]/Ni 交易/v 所持有的/d [slot2]/Ni 份额/n

[slot1]/Ni 减持/v 所持有的/d [slot2]/Ni 份额/n

cost: 0 + 10 + 0 + 0 + 0 =10 < Threshold

2. Event extracting

Bootstrapping-pattern filtration

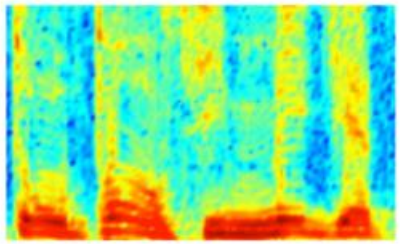
*[slot1]/Ni */v 所持有的/d *[slot2]/Ni 份额/n

Filter rules:

- No entities.
- Same as existing pattern
- Only ‘*’ between [slot]

Vector Representations of Words

AUDIO



Audio Spectrogram

DENSE

IMAGES

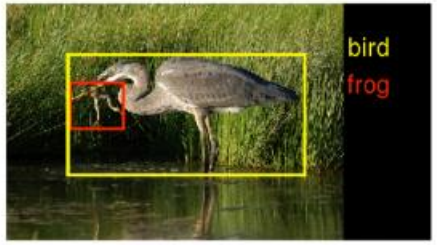


Image pixels

DENSE

TEXT

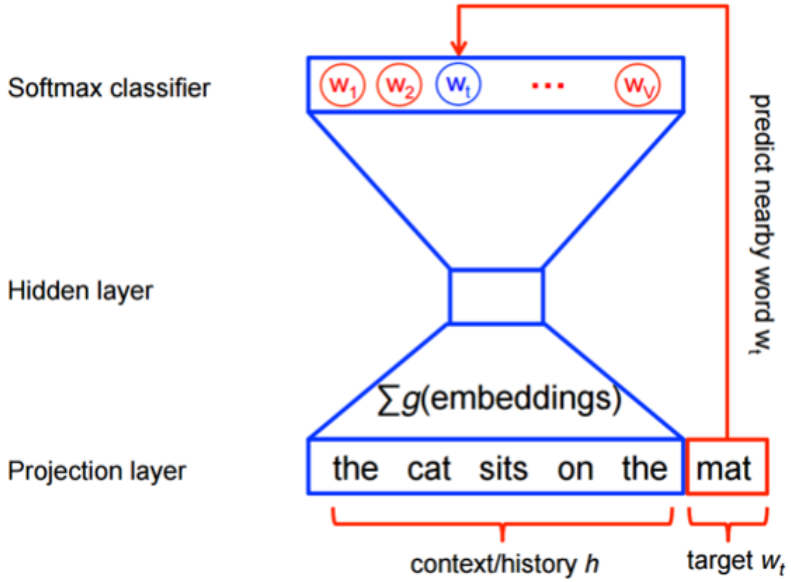
0	0	0	0.2	0	0.7	0	0	0	...	...

Word, context, or document vectors

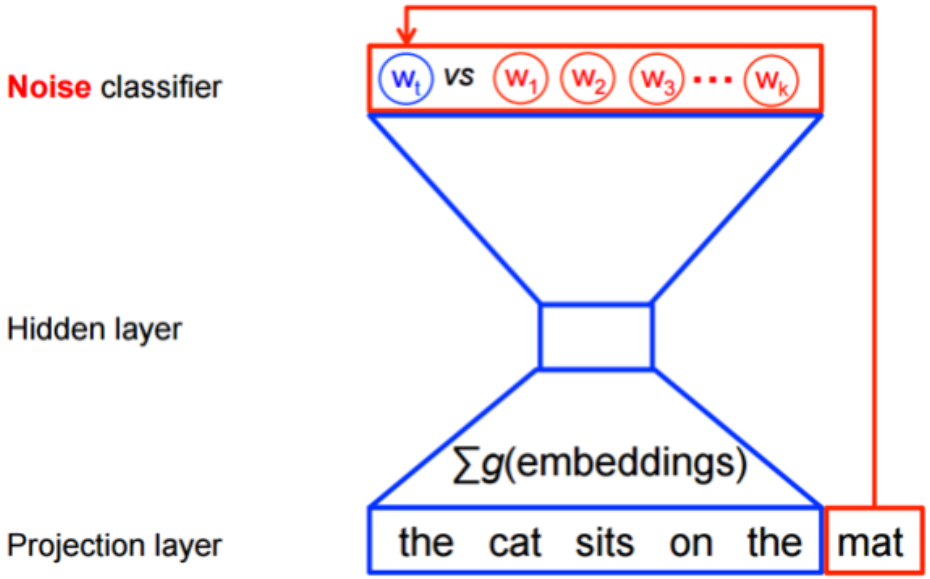
SPARSE

3. Event embedding

Neural Probability Language Model



Continue Bag-Of-Word(CBOW)



3. Event embedding

Skip-gram model

the quick brown fox jumped over the lazy dog

Get the dataset (context, target) with a window size of 1

([the, brown], quick), ([quick, fox], brown), ([brown, jumped], fox), ...

Skip-gram model inverts the contexts and targets, so the train set is

(quick, the), (quick, brown), (brown, quick), (brown, fox), ...

update to the embedding parameters θ to maximize this objective function

$$J_{NEG}^{(t)} = \log Q_\theta(D = 1|the, quick) + \log Q_\theta(D = 0|sleep, quick)$$

3. Event embedding

neural tensor network

The model returns a high score if they are in that relationship and a low one otherwise.

$$g(e_1, R, e_2) = u_R^T f\left(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R\right)$$

Where $f = tanh,\ e_1, e_2 \in R^d$ and $W_R^{[1:k]} \in R^{d \times d \times k}, e_1^T W_R^{[1:k]} e_2$ returns a vector $h \in R^k, V_R \in R^{k \times 2d}, u_R\ and\ b_R \in R^k$

$$J(\theta) = \sum_{i=1}^{N} \sum_{c=1}^{C} \max\left(0, 1 - g\left(T^{(i)}\right) + g\left(T_c^{(i)}\right)\right) + \alpha \|\theta\|_2^2$$

Where $T = (e_1, R, e_2)\ and\ T_c = (e_1, R, e_c), e_c$ is sampled randomly from the set of all entities

3. Event embedding

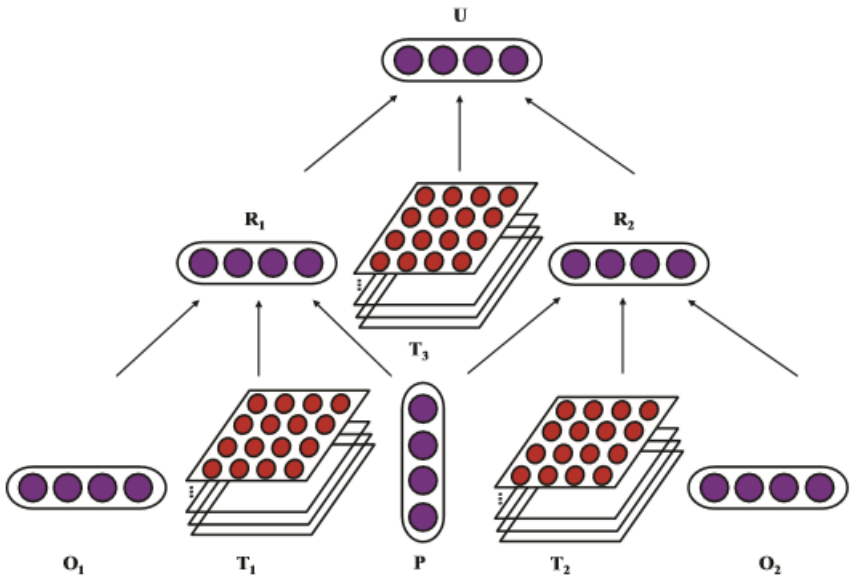
neural tensor network

d*d

$$e_1^T W_R^{[1:k]} e_2 = \begin{bmatrix} e_1^T \\ e_1^T \\ e_1^T \\ e_1^T \\ \dots \end{bmatrix} \quad \begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix} \quad \begin{bmatrix} e_2 \\ e_2 \\ e_2 \\ e_2 \\ \dots \end{bmatrix} \quad k$$

3. Event embedding

neural tensor network



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

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 $EELM'$

1 random replace the event argument and got the corrupted event tuple
2 $\mathcal{E}^r \leftarrow (E_1^r, E_2^r, \cdots, E_n^r)$
3 **while** $\mathcal{E} \neq [\,]$ **do**
4 $\quad loss \leftarrow max(0, 1 - f(E_i) + f(E_i^r)) + \lambda \|\Phi\|_2^2$
5 $\quad$ **if** $loss > 0$ **then**
6 $\quad\quad Update(\Phi)$
7 $\quad$ **else**
8 $\quad\quad \mathcal{E} \leftarrow \mathcal{E} / \{E_i\}$
9 **return** $EELM$

A score or distance?

4. Future work

1. Refine event extracting
2. Bootstrapping
3. Design a score or distance

Reference

1. Ding X, Zhang Y, Liu T, et al. Deep learning for event-driven stock prediction[C]. international joint conference on artificial intelligence, 2015: 2327-2333.
2. Fader A, Soderland S, Etzioni O, et al. Identifying relations for open information extraction[C]. empirical methods in natural language processing, 2011: 1535-1545.
3. https://github.com/HIT-SCIR/pyltp
4. Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[C]. neural information processing systems, 2013: 926-934.
5. https://www.tensorflow.org/