A Neural Model for Joint Event Detection and Summarization

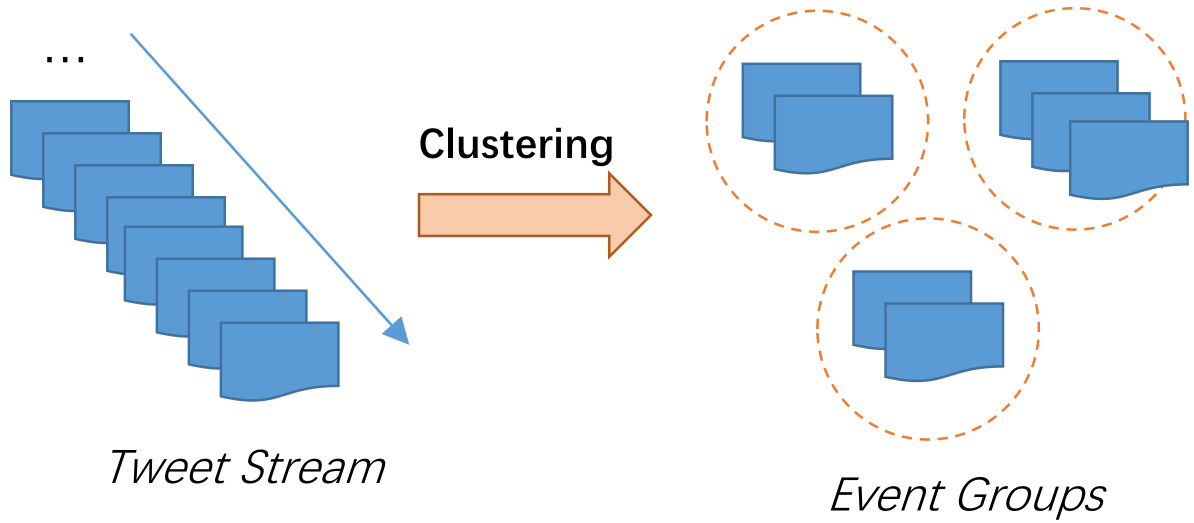
Zhongqing Wang[1,2] and Yue Zhang[2]

[1]Soochow University, China

[2]Singapore University of Technology and Design

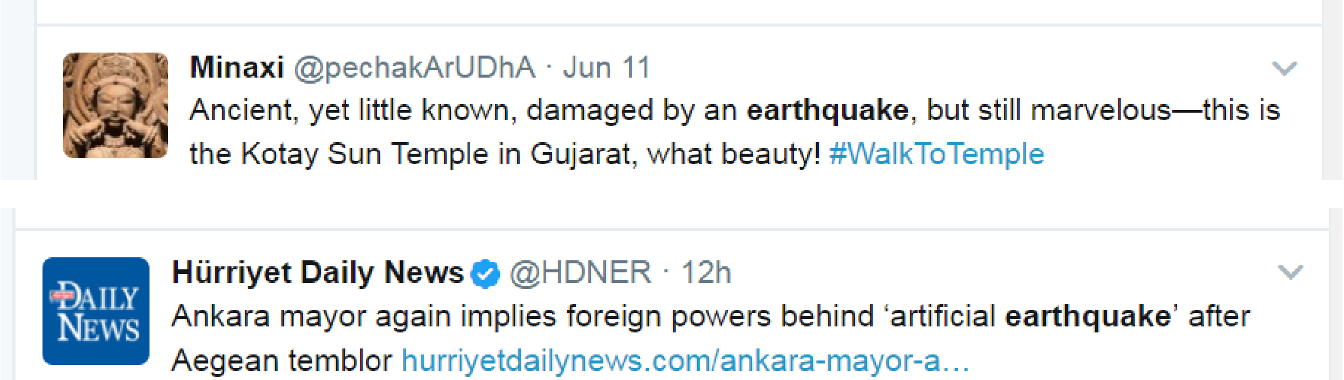
Tweet New Event Detection

- Aims to identify **first stories** in a tweet stream
  - *Incremental clustering* is always used to cluster tweets into event groups.



Challenges of Tweet New Event Detection

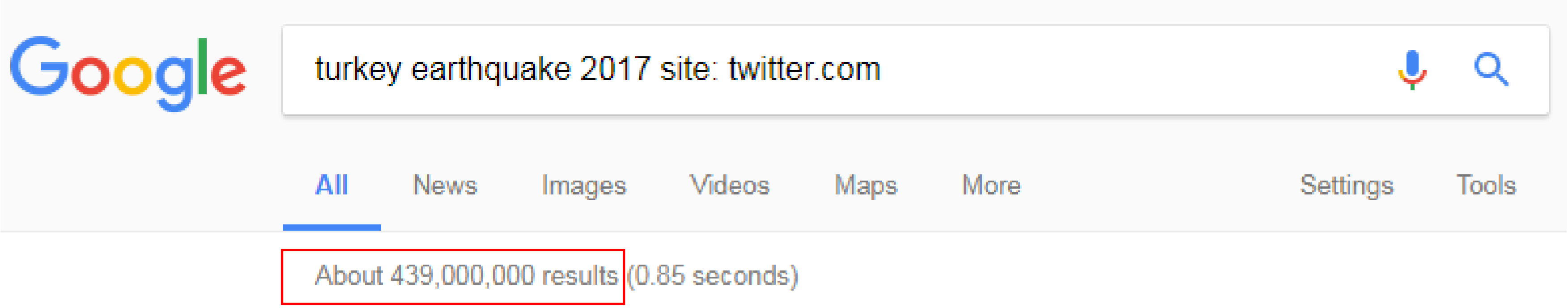
- There are lots of noise tweets in the tweet stream



Contain earthquake keyword. But do not mention any earthquake event

Need to filter

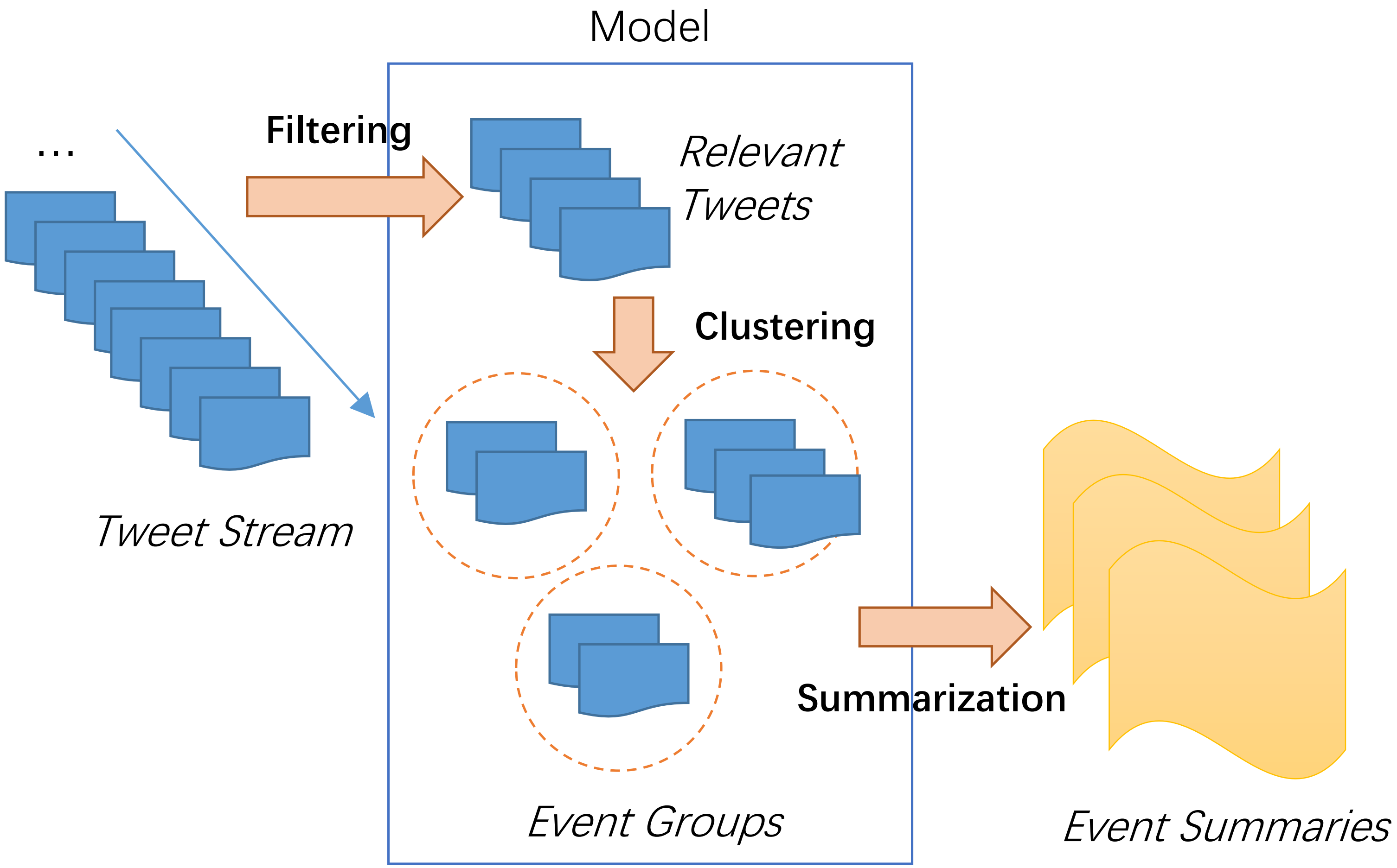
- Some events are mentioned by too many tweets



Need to summarize

Solution

- Not only cluster events, but also filter tweets and summarize events.
  - Tweets Filtering
  - Event Clustering
  - Event Summarization



Correlation between Different Stages

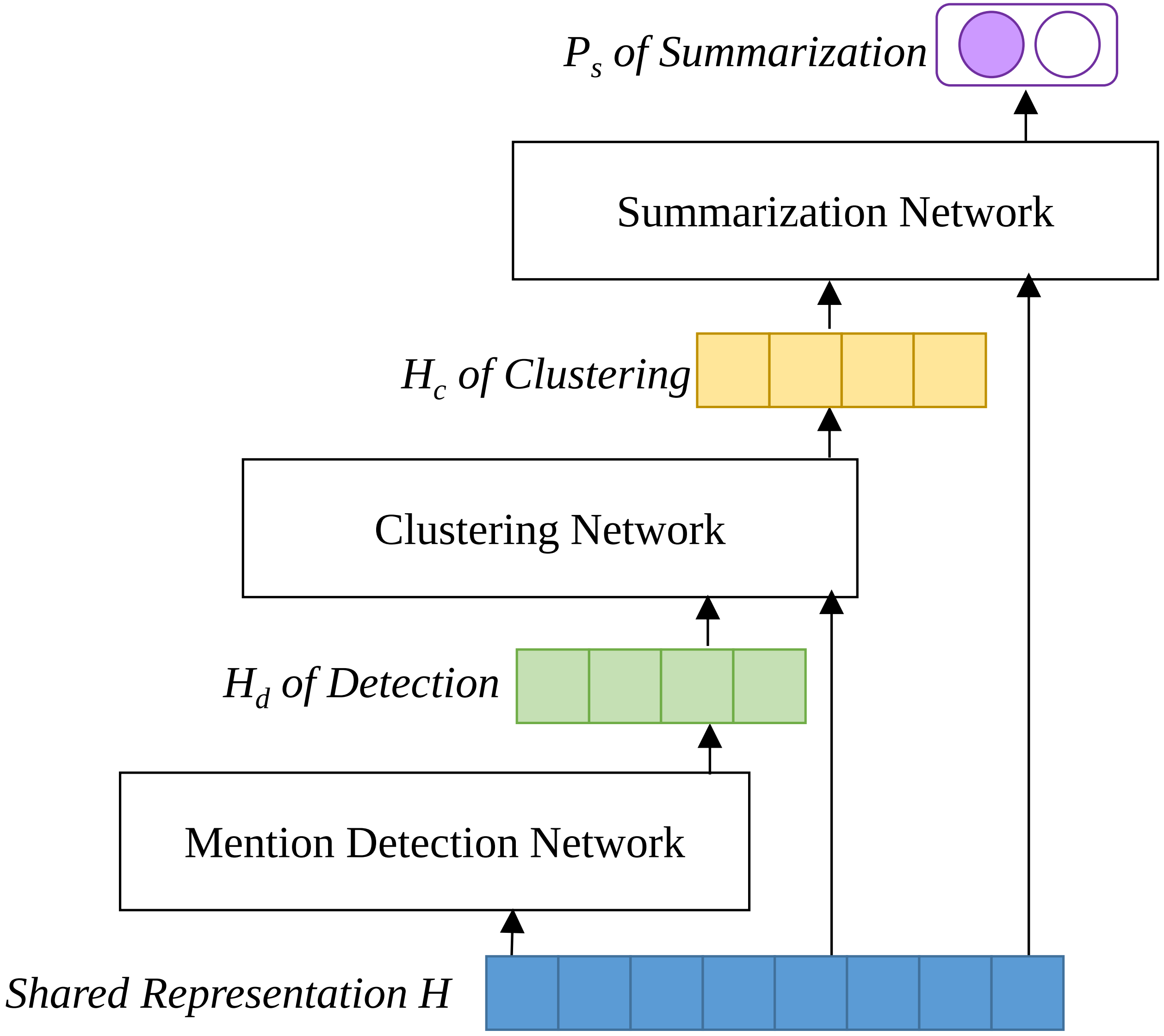
- A tweet that comprehensively describes an event should be scored highly in both the *relevance-filtering* and the *extractive-summarization* steps.
- Better understanding of a tweet is helpful for both *relevance-filtering* and *event-clustering*.

Detect and Summarize Event Jointly

- A deep neural network is used to model the three subtasks jointly
  - *Representation learning* is used to transform each incoming tweet into a dense low dimension vector
  - *Neural stacking* is used to integrate different subtasks.

Overview of Joint Event Detection and Summarization

- Shared Representation
  - LSTM
- Joint Model
  - Tweet Filtering
  - Event Clustering
  - Event Summarization



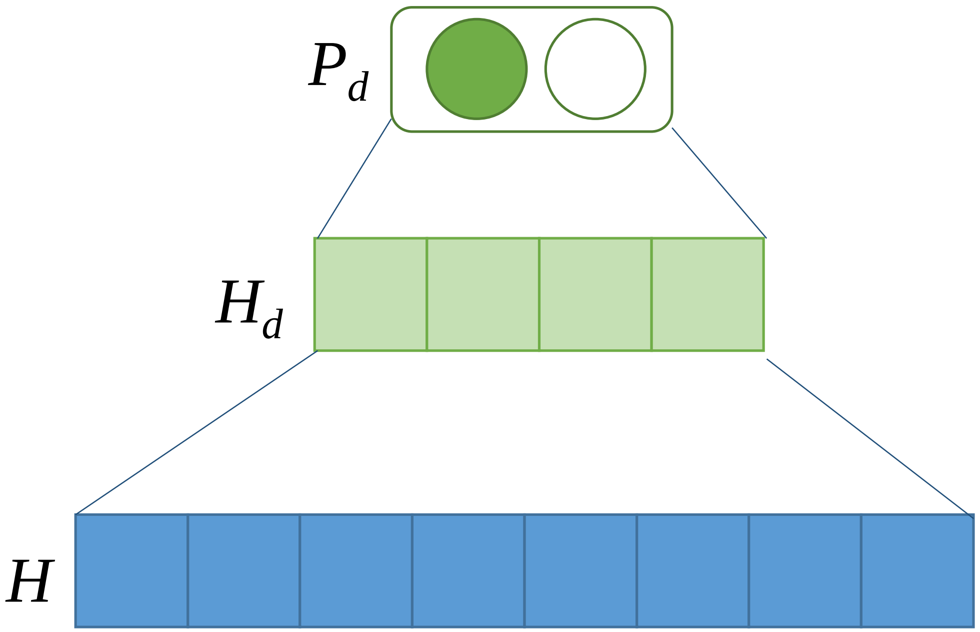
Tweet Filtering

- We classify each tweet in the stream as either being relevant or irrelevant to the events of concern.
  - A binary classification task
  - A multi-layer perceptron

$$H_d = \sigma(W_d^h \boxed{H} + b_d^h),$$

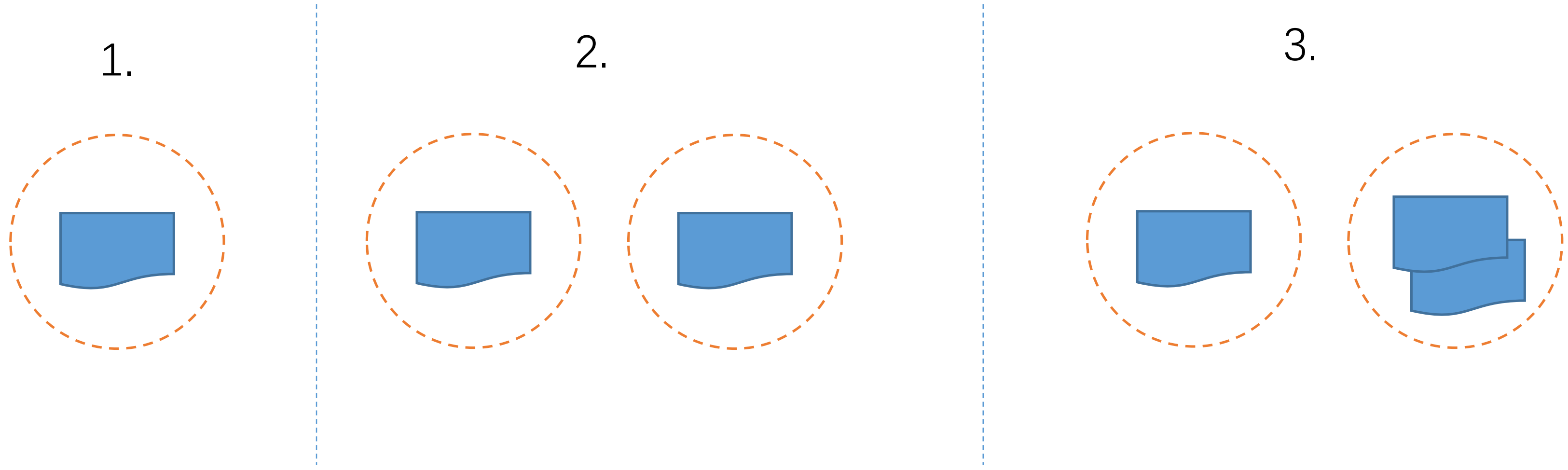
hidden variables of tweet

$$P_d = \text{softmax}(W_d H_d + B_d)$$



Event Clustering

- Incremental clustering of tweets [Aggarwal and Subbian, 2012].
  - Given a new tweet, decide whether it belongs to an existing event cluster, or describes a new event
  - A key issue is the calculation of *similarity between tweets*.



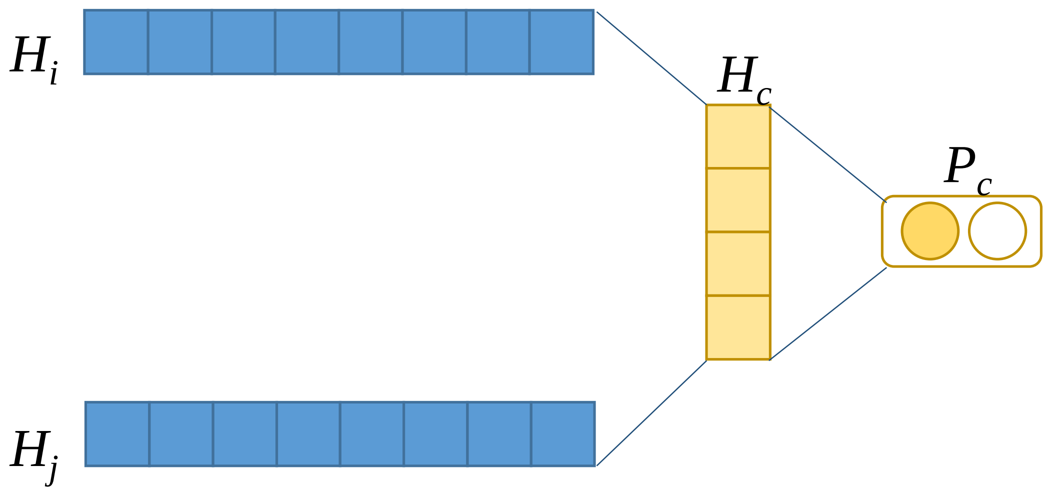
Siamese Network for calculating similarity

- Siamese Network

$$H_c = \sigma(W_c^h(\boxed{H_i \oplus H_j}) + b_c^h)$$

hidden variables of tweet *i* and *j*

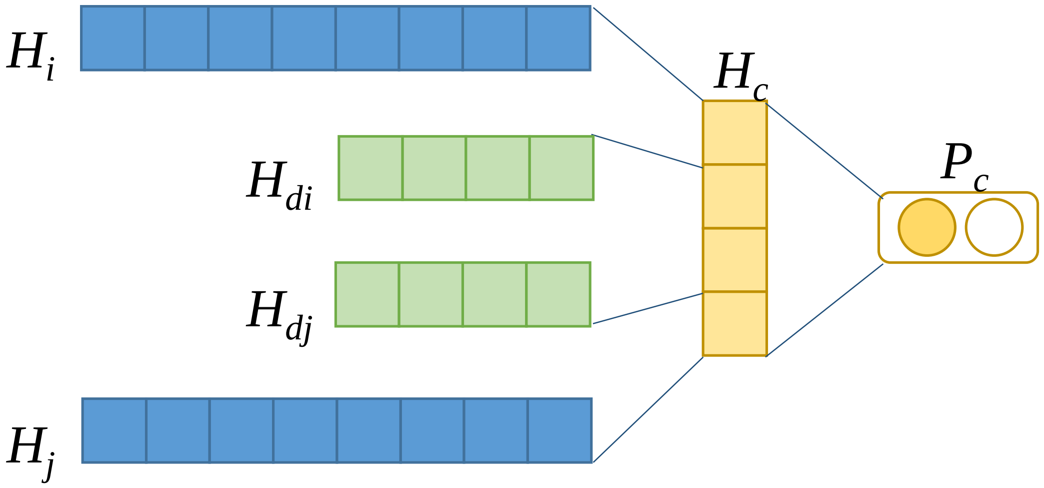
$$P_c = \mathrm{softmax}(W_c H_c + B_c)$$



Integrating with tweet filtering

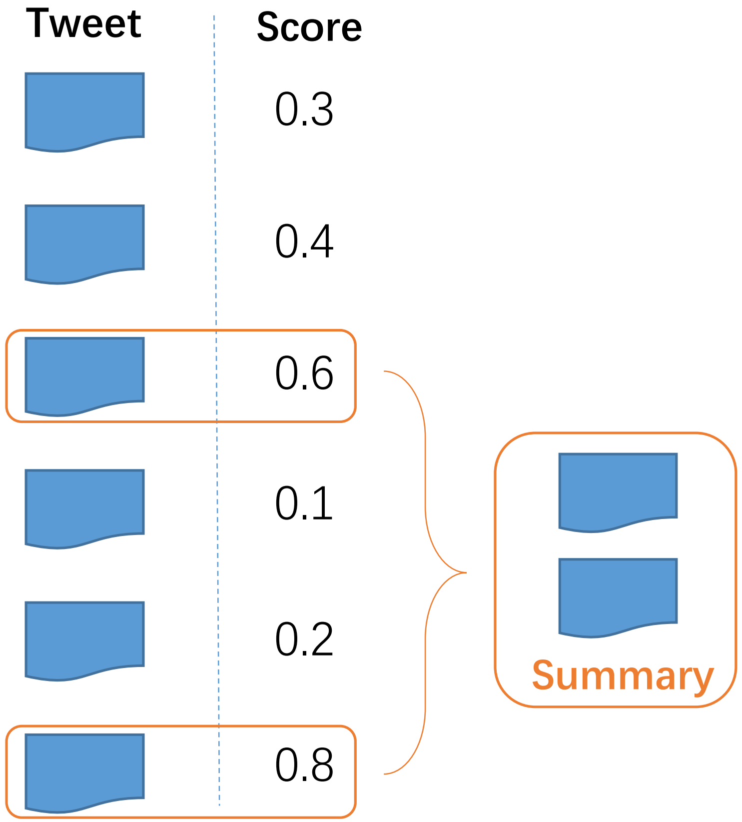
$$H_c = \sigma(W_c^h(H_i \oplus H_j) + b_c^h) \Rightarrow H_c = \sigma(W_c^h(H_i \oplus H_j \oplus \boxed{H_{d_i} \oplus H_{d_j}}) + b_c^h),$$

Hidden variables from tweet filtering



Event Summarization

- We rank all the tweets in the cluster using a probability score, and select top-n to build the summary.



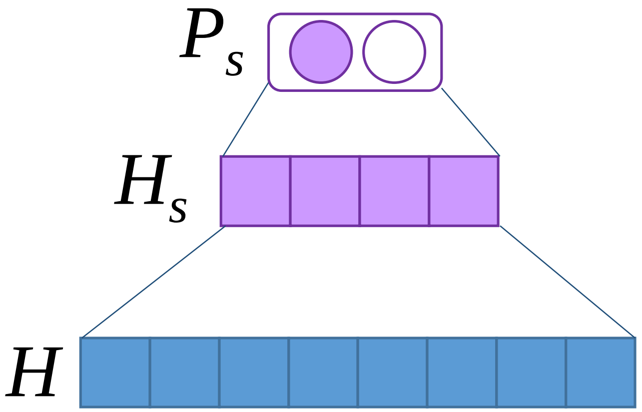
Event Summarization (cont.)

- A multi-layer perceptron

$$H_s = \sigma(W_s^h \boxed{H} + b_s^h)$$

hidden variables of tweet

$$P_s = \mathrm{softmax}(W_s H_s + B_s)$$

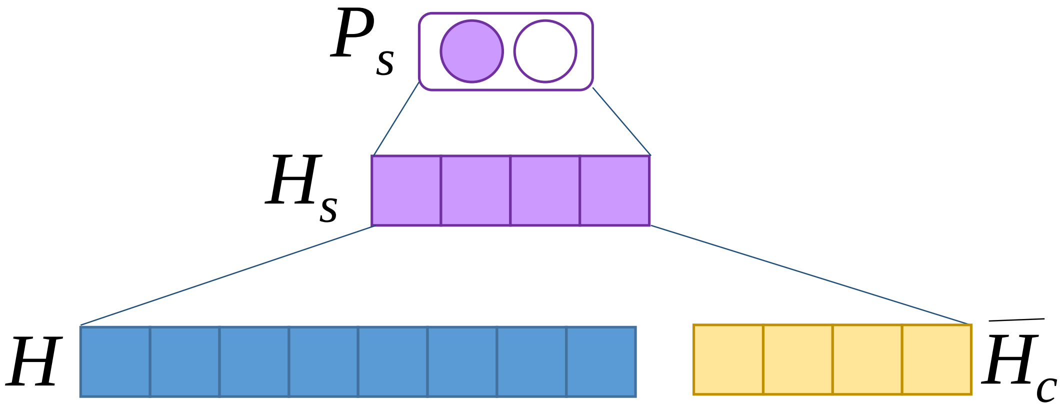


Integrating with event clustering

$$H_s = \sigma(W_s^h(H \oplus \boxed{\overline{H_c^h}}) + b_s^h)$$

$$P_s = \mathrm{softmax}(W_s H_s + B_s)$$

- $\overline{H_c^h}$ is the sum of H_c^h between the tweet X and all the other tweets in the same cluster



Data Collection

- All data were collected by using the Twitter streaming API
  - consist of tweets from June 2013 until April 2016
- The tweets are collected with relevant domain keywords
  - Earthquake:
    - earthquake, shake, refugees, victims
  - DDoS:
    - ddos, anonymous attack, spoofed attack, zombies host

Event Annotation

- We adopt the approach employed by NIST in labeling TDT data [Allan, 2002]
  - A relevant tweet must explicitly mention the event
  - The main purpose of the tweet should be to inform of the event
- Statistic of dataset

	Earthquake	DDoS
#Event	47	170
#Post	12090	17760
Vocabulary size	11462	15032

Evaluation Metrics

- Clustering
  - We use the standard TDT evaluation procedure [Allan, 2002], where normalized *Topic Weighted Minimum Cost* (C_{min}) is taken for evaluating clustering accuracy
- Summarization
  - We use ROUGE-1.5.5 [Lin, 2004] for summary evaluation. We report *unigram overlap (ROUGE-1)* for assessing informativeness.
- We evaluate our proposed model and analyze the influence of different factors on *earthquake* domain.

Effectiveness of Event Mention Detection

- The event clustering performance with/without the event mention detection.
  - *Cosine* is a traditional strategy with bag-of-words as document representation [Aggarwal and Subbian, 2012]
  - *LSTM* means calculating the similarity using the LSTM based Siamese network [Mueller and Thyagarajan, 2016].

Method	C_{min}
Random	86.2
Cosine − filtering	65.8
Cosine + filtering	60.9
LSTM − filtering	64.4
LSTM + filtering	58.8

Event filtering always outperform those without event mention filtering

Neural Network is better than discrete model

Effectiveness of Joint Modeling

- The results of different ablation baselines

Method	Clustering	Summarization
LSTM-Pipeline	58.8	18.2
LSTM-Joint	52.2	19.4
+Detect	50.2	20.6
+Cluster	47.2	20.1
JEDS	**45.8**	**21.3**

Only integrate *filtering* for *clustering*

Only integrate *clustering* for *summarization*

Comparison with State-of-the-art

- Comparison of clustering algorithms

State-of-the-art models for event clustering

Method	C_{min}
LSH	66.7
AS12	60.9
JEDS	**45.8**

- Comparison of summarization algorithms

State-of-the-art model for event clustering and summarization →

Method	ROUGE-1
AS12+LexRank	18.8
AS12+CL16	19.6
LSH+LexRank	17.2
LSH+CL16	19.1
JEDS	**21.3**

Results on DDoS Domain

- Comparison with state-of-the-art

Method	Clustering	Summarization
AS12+LexRank	64.4	15.5
LSH+CL16	57.8	16.5
JEDS	**38.3**	**18.7**

Thanks

wangzq.antony@gmail.com, yue_zhang@sutd.edu.sg