Early Rumour Detection

Kaimin Zhou[2] Chang Shu[1,2] Binyang Li[3] **Jey Han Lau**[4,5]

[1] School of Computer Science,
University of Nottingham Ningbo China

[2] DeepBrain

[3] School of Information Science and Technology,
University of International Relations

[4] School of Computing and Information Systems,
The University of Melbourne

[5] IBM Research Australia

will@deepbrain.ai, scxcs1@nottingham.edu.cn,
byli@uir.edu.cn, jeyhan.lau@gmail.com

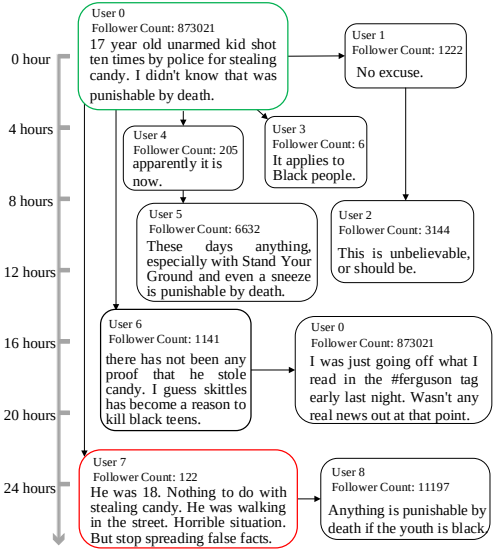
Introduction

- Rumours can spread quickly through social media.
- Malicious ones can bring about significant economical and social impact.
- Our paper focuses on the task of rumour detection.
- Particularly, we are interested in understanding how *early* we can detect them.

Importance of Timeliness

- Rumour detection isn't a new task: there are numerous studies and data sets on rumour detection.
- Few, however, are concerned with the timing of the detection.
- A successfully-detected malicious rumour can still cause significant damage if it isn't detected in a timely manner.
- Timing is crucial.

Michael Brown's Shooting on Twitter



How Rumour Propagates

- Source message (green box) started a claim about the cause of Michael Brown's shooting.
- It claimed that he was shot for stealing candy.
- The dramatic claim was retweeted by several influential users, and within 24 hours about 900K users were involved.
- Only after 24 hours we see a user (red box) questioned the veracity of the source message.
- Had the rumour been identified earlier and rebutted, its propagation could have been contained.

Background

- Most studies (Qazvinian et al. [2011], Zhang et al. [2015]) consider rumour detection as a binary classification problem.
- More recent works (Long et al. [2017], Ruchansky et al. [2017]) explore deep learning methods to enhance detection accuracy.
- In all these studies, however, timeliness isn't evaluated.
- There are a few exceptions, e.g. Ma et al. [2015] and Kwon et al. [2017].
- In these papers, the authors define a checkpoint in the timeline and use all posts prior to the checkpoint to classify a rumour.
- Checkpoint is a pre-determined value (e.g. after N posts), and so does not capture the variation of propagation patterns for different rumours.

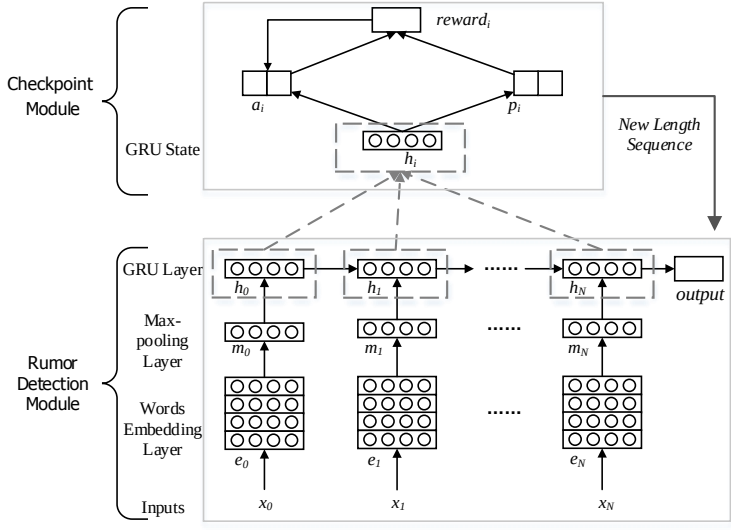
Our Approach

- We combine deep learning and reinforcement learning to identify rumours as early as possible.
- Our early rumour detection system (ERD) features two modules:
  - A rumour detection module (RDM) that classifies whether an event constitutes a rumour;
  - A checkpoint module (CM) that determines when to trigger RDM.
- What is an event? It's a collection of posts consisting a source message and all responses and reposts

High Level Description of ERD

- ERD treats incoming posts as a data stream.
- When a new post arrives, this post (along with all related prior posts) will be used to decide if it constitutes an appropriate checkpoint to trigger RDM.
- ERD integrates reinforcement learning for CM to guide RDM, using RDM's classification accuracy as a reward.
- Through this, ERD is able to learn the minimum number of posts required to identify a rumour.
- In other words, checkpoint in ERD is *dynamic*, and that's the core novelty of our methodology.

Model Architecture



Rumour Detection Module (RDM)

- Consists of several layers:
  - Word Embedding: maps input words into vectors
    $x_i \rightarrow [e_i^0; e_i^1; ...; e_i^K]$
  - Max-pooling layer: extract salient features for a post
    $m_i = \text{maxpool}([\mathbf{W}_m e_i^0; \mathbf{W}_m e_i^1; ...; \mathbf{W}_m e_i^K])$
  - GRU: capture temporal relationship between multiple posts
    $h_i = \text{GRU}(m_i, h_{i-1})$
  - Output layer:
    $p = \text{softmax}(\mathbf{W}_p h_N + b_p)$
    where $N =$ number of posts received to date, and $p \in \mathbb{R}^2$, i.e. p^0 (p^1) gives the probability of the positive (negative) class

Checkpoint Module (CM)

- CM uses deep Q-learning model (Mnih et al. [2013]).
- The optimal action-value function $Q^*(s, a)$ is defined as the maximum expected return achievable under state s:

$$Q^*(s, a) = E_{s' \ \varepsilon}[r + \gamma \max_{a'} Q_i(s', a')|s, a]$$

where r is the reward value, γ the discount rate.

- To compute the action-value function, we use the hidden states produced by the GRU in RDM:

$$a_i = \mathbf{W}_a(\text{ReLu}(\mathbf{W}_h h_i + b_h)) + b_a$$

where $a_i \in \mathbb{R}^2$ is the action value for *terminate* (a_i^0) or *continue* (a_i^1) at post x_i.

Joint Training

- Training process is similar to that of generative adversarial networks (Goodfellow et al. [2014]).
- Key contrast: RDM and CM is working cooperatively rather than adversarially.
- We pre-train RDM based on cross-entropy before joint training.
- During joint training we train CM and RDM in an alternating fashion.

Reward for CM

- In each step of the training, new posts will be processed by RDM (to generate the hidden states h_i) which will in turn be used by CM to calculate the action values (a_i).
- If the system takes the *terminate* action, the reward is given based on RDM's prediction; otherwise a small penalty is incurred:

$$r_i = \begin{cases} \log M, & \textit{terminate} \text{ with correct prediction} \\ -P, & \textit{terminate} \text{ with incorrect prediction} \\ -\varepsilon, & \textit{continue} \end{cases}$$

  - where M is the number of correct predictions accumulated thus far;
  - P is a large value to penalise an incorrect prediction;
  - ε is a small penalty value for delaying the detection.

Data Set

- We use two standard rumour data sets: `Weibo` (Ma et al. [2016]) and `Twitter` (Zubiaga et al. [2016]).
- 10% events reserved as validation; rest is split in a ratio of 3:1 for train and test.

Statistics	`Weibo`	`Twitter`
User#	2,746,818	49,345
Posts#	3,805,656	103,212
Events#	4,664	5,802
Rumours#	2,313	1,972
Non-rumours	2,351	3,830
Avg. hours per event	2,460.7	33.4
Avg. # of posts per event	816	17
Max # of posts per event	59,318	346
Min # of posts per event	10	1

Models

- **Baseline**: SVM with tf-idf features
- **CSI** (Ruchansky et al. [2017]): neural model that integrates text and user information to classify rumours.
- **CRF** and **HMM** (Zubiaga et al. [2016], Dungs et al. [2018]): classical models that use crowd opinions of the event for classification.
- **GRU-2** (Ma et al. [2016]): two-layer GRU with tf-idf features.
- **RNN**, **LSTM** and **GRU-1**: variants of GRU-2 with simpler recurrent architectures.

Detection Accuracy: Weibo

Method	Accuracy	Precision	Recall	F1
Baseline	0.724	0.673	0.746	0.707
RNN	0.873	0.816	0.964	0.884
LSTM	0.896	0.846	**0.968**	0.913
GRU-1	0.908	0.871	0.958	0.913
GRU-2	0.910	0.876	0.956	0.914
CSI*	0.953	—	—	0.954
RDM	**0.957**	**0.950**	0.963	**0.957**
ERD	0.933	0.929	0.936	0.932

Detection Accuracy: Twitter

Method	Accuracy	Precision	Recall	F1
Baseline	0.612	0.355	0.465	0.398
RNN	0.785	0.707	0.659	0.682
LSTM	0.796	0.719	0.683	0.701
GRU-1	0.800	0.735	0.685	0.709
GRU-2	0.808	0.741	0.694	0.717
CRF*	—	0.667	0.566	0.607
HMM*	—	—	—	0.524
RDM	**0.873**	0.817	**0.823**	**0.820**
ERD	0.858	**0.843**	0.735	0.785

Detection Accuracy: Findings

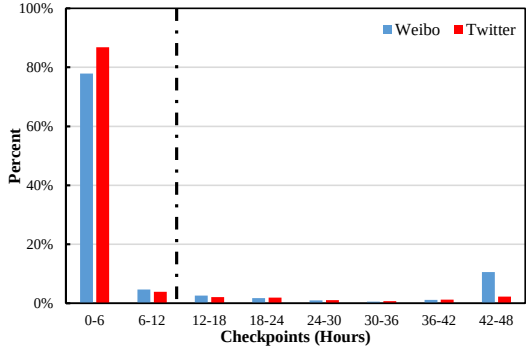
- RDM outperforms all models across most metrics.
- ERD performs very competitively, outperforming most benchmark systems and baselines, with the exception of CSI on `Weibo`.
- Unlike all other systems, ERD uses only a subset of posts (average = 4.03 posts) for rumour classification.
- Exception: HMM is the only benchmark that uses a subset (first 5 posts), but its performance is markedly worse.

Detection Timeliness

- Compare ERD against GRU-2, as it performed competitively for both data sets.
- GRU-2 uses a manually set checkpoint (12 hours after source message), which were found to be optimal.

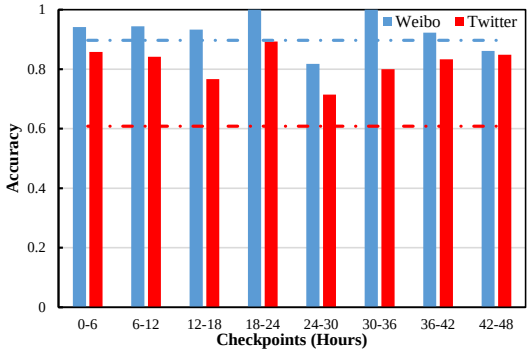
Classified Events Over Time

- We first present the proportion of events that are classified by ERD over time.
- Approximately 80% events are classified within first 6 hours.
- GRU-2's optimal checkpoint is 12 hours (dashed), so ERD's detection is earlier.



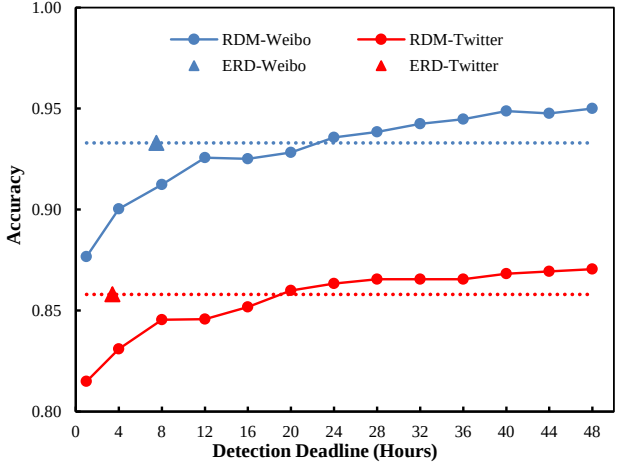
Accuracy Over Time

- We next present classification accuracy over time.
- ERD outperforms GRU-2 over all check points.
- Although checkpoints longer than 12 hours are not exactly comparable (since ERD uses more posts than GRU-2).



ERD vs. RDM

- We next compare ERD and RDM to understand the impact of CM.



Impact of CM?

- Dashed lines indicate average performance of ERD, which detects rumours on average in 7.5 and 3.4 hours on `Weibo` and `Twitter` respectively.
- Solid lines show accuracy of RDM, which increases over time as it has more evidence.
- For RDM to achieve the same performance as ERD, it requires approximately at least 20 hours of posts.
- These observations highlights the importance of the checkpoint module, which allows ERD to detect rumours much earlier.
- In certain events, they are detected within 3 minutes.

Case Study: Toxic Crabs on Weibo

- A set of salient words (2nd column) are extracted from posts published during a particular period (1st column) using tf-idf features.

Interval	Salient Words	Translation
18:41 – 18:44	大闸蟹，毒性，激素，有害，吃惊	hairy crabs, toxicity, hormone, harmful, amazed
18:48 – 18:51	大闸蟹，爆出，消息，吃惊，上市	hairy crabs, bursts, message, amazed, on the market
18:51 – 18:59	美食，为何，这样，晕，同城会	delicious food, why, so, dizzy, one city club
18:59 – 19:09	敢吃吗，吃得起，喜欢，惨，偷笑	dare to eat, afford to eat, like, miserable, laughing
19:11 – 19:15	食品安全，真的吗，失望，神马，不能	food safety, really, disappointment, what, cannot
	Rumour Detected	
19:34 – 19:49	是不是，大闸蟹，吃不成，疑问，围观	is it, hairy crabs, cannot eat, doubt, look around

Case Study: Toxic Crabs on Weibo

- The rumour was started by a message claiming that hairy crabs contain harmful toxins on August 18th, 2012.
- Within 12 hours, 2.3M users participated in its propagation.
- The rumour spread quickly and led to significant economic damage to the aquaculture industry in China.
- It was officially rebutted after 24 hours, but ERD detects the rumour in less than an hour.

Conclusion

- We present ERD, an early rumour detection system.
- ERD learns dynamically the minimum number of posts required to identify a rumour.
- ERD integrates reinforcement learning with deep learning to monitor microblogs in real time to decide when classify rumours.
- Across two data sets in different languages, ERD achieves a competitive detection accuracy compared to state-of-the-art systems.
- In terms of detection timeliness, ERD identifies rumours much earlier: on average 4.5 or 8.6 hours earlier depending on the dataset.

Questions?

References I

Sebastian Dungs, Ahmet Aker, Norbert Fuhr, and Kalina Bontcheva. Can rumour stance alone predict veracity? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3360–3370, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. Rumor detection over varying time windows. *PloS one*, 12(1):e0168344, 2017.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. Fake news detection through multi-perspective speaker profiles. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 252–256, 2017.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754. ACM, 2015.

References II

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824, 2016.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics, 2011.

Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806. ACM, 2017.

Qiao Zhang, Shuiyuan Zhang, Jian Dong, Jinhua Xiong, and Xueqi Cheng. Automatic detection of rumor on social network. In *Natural Language Processing and Chinese Computing*, pages 113–122. Springer, 2015.

References III

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363*, 2016.