Rumor Detection on Twitter with Tree-structured Recursive Neural Networks

Jing Ma[1], Wei Gao[2], Kam-Fai Wong[1,3]
[1]The Chinese University of Hong Kong
[2]Victoria University of Wellington, New Zealand
[3]MoE Key Laboratory of High Confidence Software Technologies, China

July 15-20, 2018 – ACL 2018 @ Melboume, Australia

Outline

- Introduction
- Related Work
- Problem Statement
- RvNN-based Rumor Detection
- Evaluation
- Conclusion and Future Work

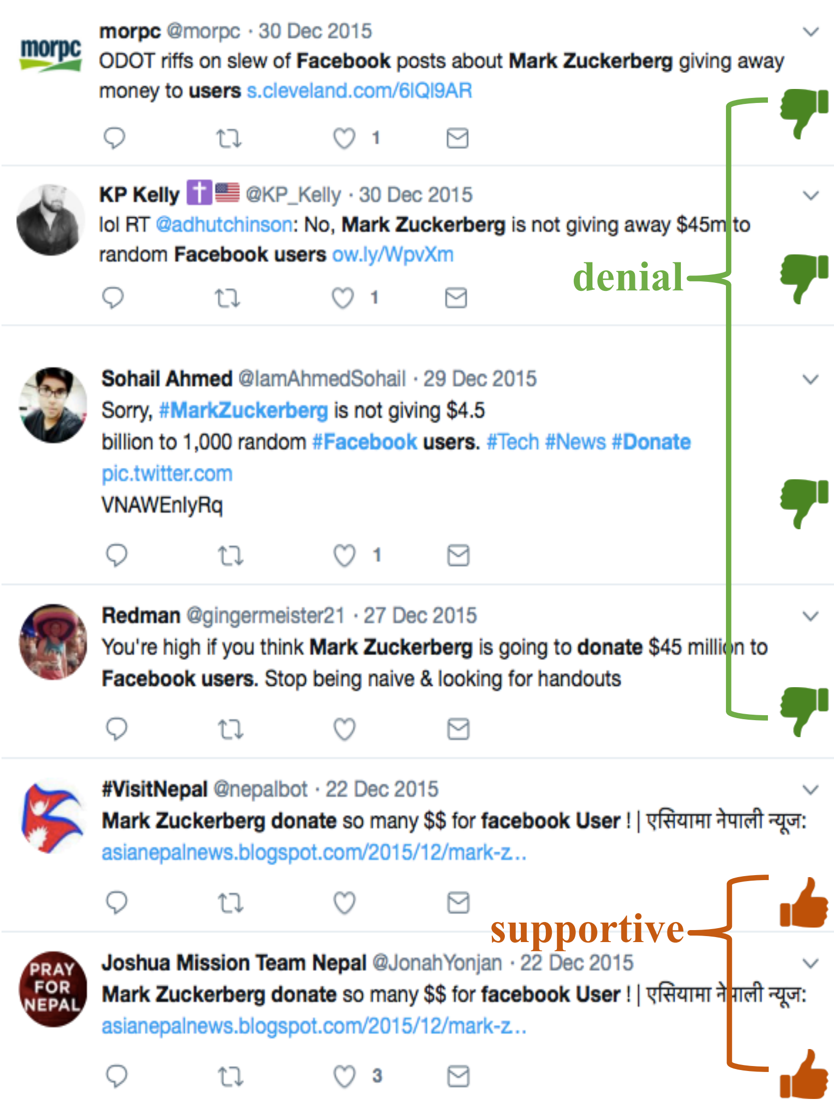
Introduction

What are rumors?

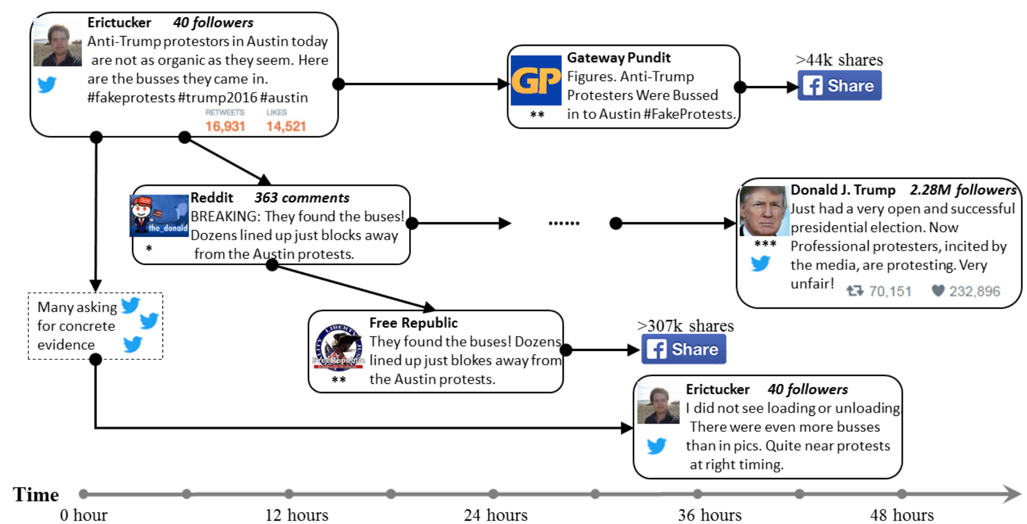
A story or statement whose truth value is **unverified** or deliberately **false**

Introduction

How the fake news propagated?



- ➢ people tend to stop spreading a rumor if it is known as false. (Zubiaga et al., 2016b)
- ➢ Previous studies focused on text mining from **sequential** microblog streams, we want to bridge the **content** semantics and **propagation** clues.

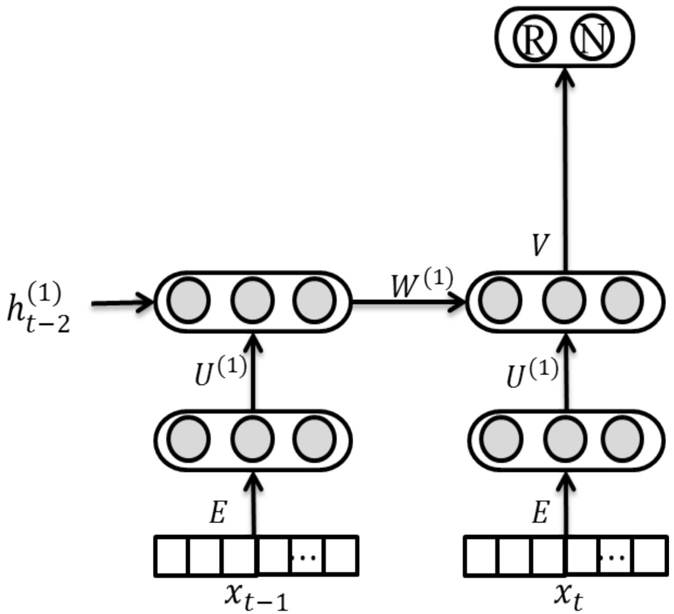


Motivation

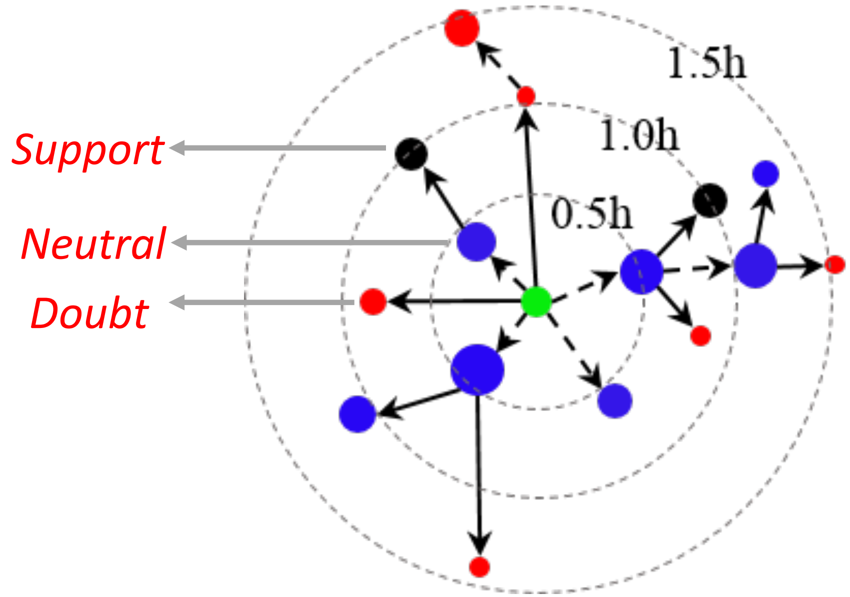
- We generally are not good at distinguishing rumors
- It is crucial to track and debunk rumors early to minimize their harmful effects.
- Online fact-checking services have limited topical coverage and long delay.
- Existing models use feature engineering – over simplistic; or recently deep neural networks – ignore propagation structures; Kernel-based method – develop based on tree structure but cannot learn high-level feature representations automatically.

Observation & Hypothesis

- Existing works: Consider ***post representation*** or ***propagation structure***



(a) RNN-based model
(Ma et al. 2016)

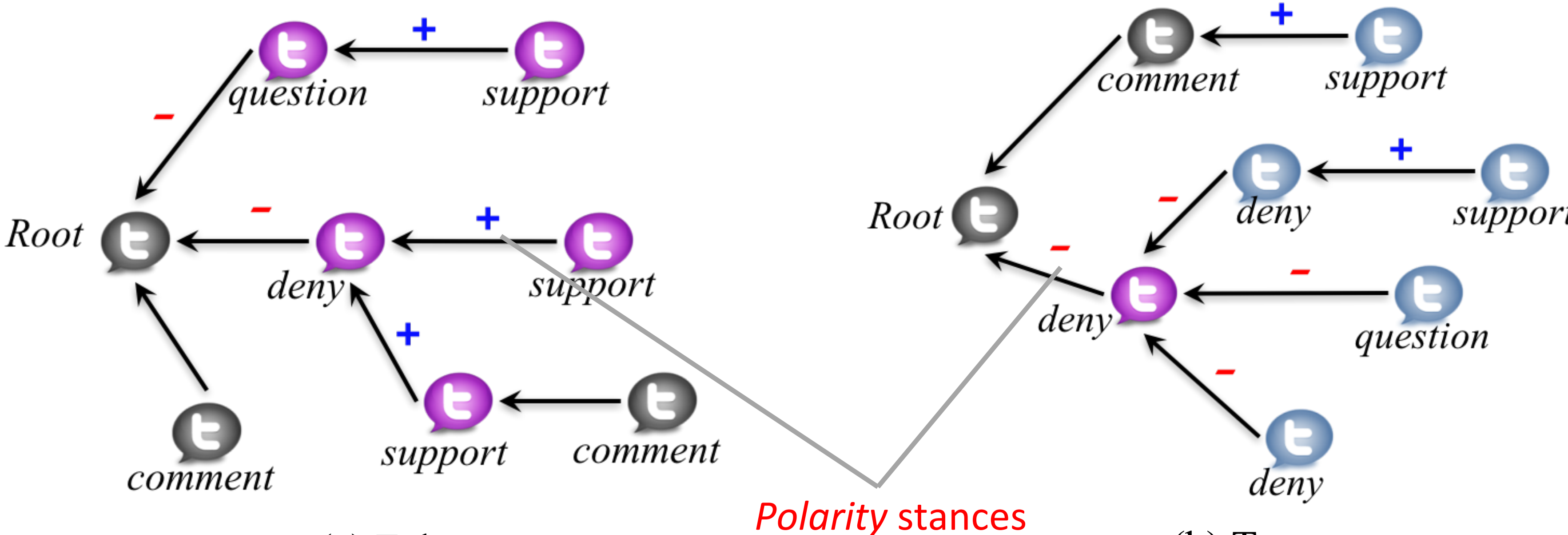


(b) Tree kernel-based model
(Ma et al. 2017)

- IDEA: Combining the two models, leveraging propagation structure by representation learning algorithm

Observation & Hypothesis

Why such model do better?



(a) False rumor

(b) True rumor

Local characteristic:

- A reply usually respond to its **immediate** ancestor rather than the **<u>root tweet</u>**.
- Repliers tend to disagree with (or question) who support a false rumor or deny a true rumor; repliers tend to agree with who deny a false rumor or support a true rumor.

Contributions

- The first study that deeply integrates both structure and content semantics based on tree-structured recursive neural networks for detecting rumors from microblog posts

- Propose two variants of RvNN models based on bottom-up and top-down tree structures, to generate better integrated representations for a claim by capturing both structural and textural properties signaling rumors.

- Our experiments based on two real-world Twitter datasets achieve superior improvements over state-of-the-art baselines on both rumor classification and early detection tasks.

- We make the source codes in our experiments publicly accessible at https://github.com/majingCUHK/Rumor_RvNN

Outline

- Introduction
- Related Work
- Problem Statement
- RvNN-based Rumor Detection
- Evaluation
- Conclusion and Future Work

Related Work

- Systems based on common sense and investigative journalism, e.g.,
  - *snopes.com*
  - *factcheck.org*
- Learning-based models for rumor detection
  - Information credibility: Castillo et al. (2011), Yang et al. (2012)
  - Using handcrafted and temporal features: Liu et al. (2015), Ma et al. (2015), Kwon et al. (2013, 2017)
  - Using cue terms: Zhao et al. (2015)
  - Using recurrent neural networks: Ma et al. (2016, 2018)
  - Tree-kernel-based model: Ma et al. (2017), Wu et al. (2015)

Without hand-crafted features

- RvNN-based works
  - images segmentation (Socher et al, 2011)
  - phrase representation from word vectors (Socher et al, 2012)
  - Sentiment analysis (Socher et al, 2013)
  - etc

Outline

- Introduction
- Related Work
- Problem Statement
- RvNN-based Rumor Detection
- Evaluation
- Conclusion and Future Work

Problem Statement

- Given a set of microblog posts $\mathrm{R} = \{r\}$, model each source tweet as a tree structure $\mathrm{T}(r) = < V, E >$, where each node v provide the text content of each post. And E is directed edges corresponding to response relation.

- Task 1 – finer-grained classification for each source post

  false rumor, true rumor, non-rumor, unverified rumor

- Task 2 – detect rumor as early as possible

Tweet Structure

bottom-up tree

x_1: #Walmart donates $10,000 to #DarrenWilson fund to continue police racial profiling…

x_2: 1:30 Idc if they killed a mf foreal. Ima always shop with @Walmart. I'm just bein honest 💁

x_3: NEED SOURCE. have a feeling this is just hearsay ...

x_4: I agree. I have been hearing this all day but no source 1:12

x_5: Exactly, i don't think Wal-Mart would let everyone know this if they did!! 2:21

top-down tree

x_1: #Walmart donates $10,000 to #DarrenWilson fund to continue police racial profiling…

x_2: 1:30 Idc if they killed a mf foreal. Ima always shop with @Walmart. I'm just bein honest 💁

x_3: NEED SOURCE. have a feeling this is just hearsay ...

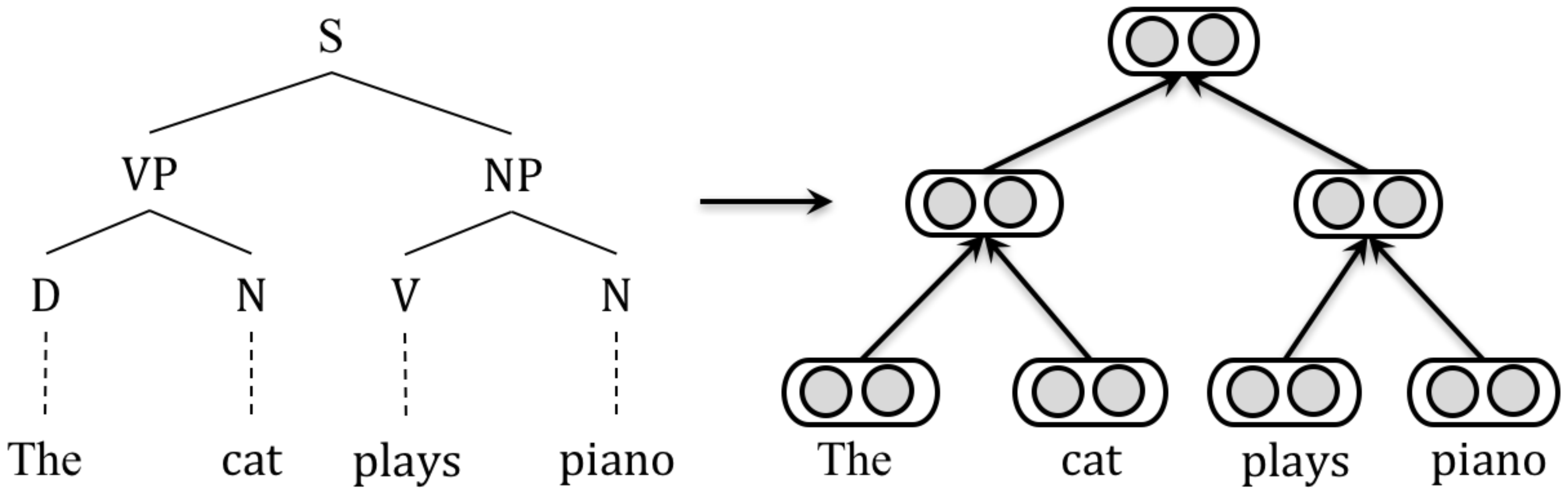
x_4: I agree. I have been hearing this all day but no source 1:12

x_5: Exactly, i don't think Wal-Mart would let everyone know this if they did!! 2:21

Outline

- Introduction
- Related Work
- Problem Statement
- RvNN-based Rumor Detection
- Evaluation
- Conclusion and Future Work

Standard Recursive Neural Networks



- RvNN (*tree-structured neural networks*) utilize sentence parse trees: representation associated with each node of a parse tree is computed from its direct children, computed by

$$p = f(W \cdot [c_1; c_2] + b)$$

  - p: the feature vector of a parent node whose children are c_1 and c_2
  - computation is done recursively over all tree nodes

Bottom-up RvNN

- **Input:** bottom-up tree (node: a post represented as a vector of words)
- **Structure**: recursively visit every node from the leaves at the bottom to the root at the top (a natural extension to the original RvNN)
- **Intuition**: local rumor indicative features are aggregated along different branches (e.g., subtrees having a denial parent and a set of supportive children) (generate a feature vector for each subtree)

- GRU equation at node *j*

$$\tilde{x}_j = x_j E$$

$$h_{\mathcal{S}} = \sum_{s \in \mathcal{S}(j)} h_s$$

Children node

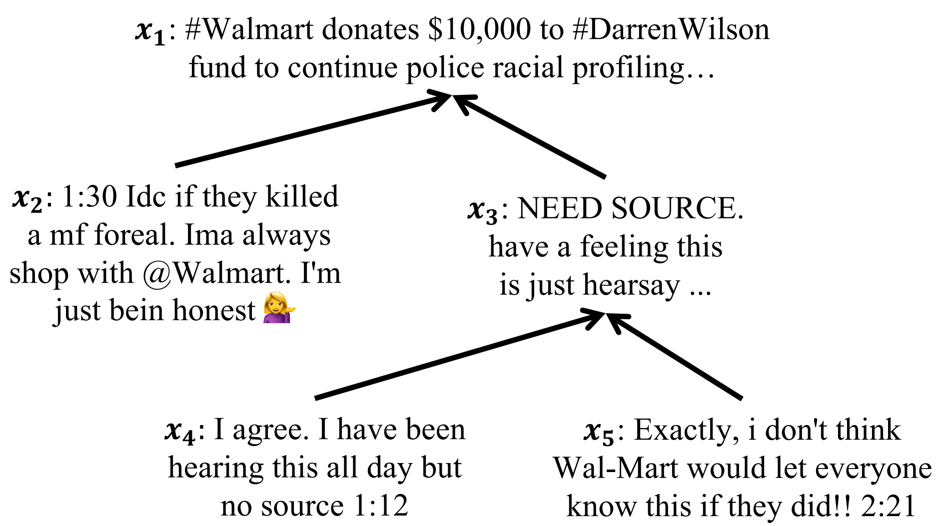
Own input

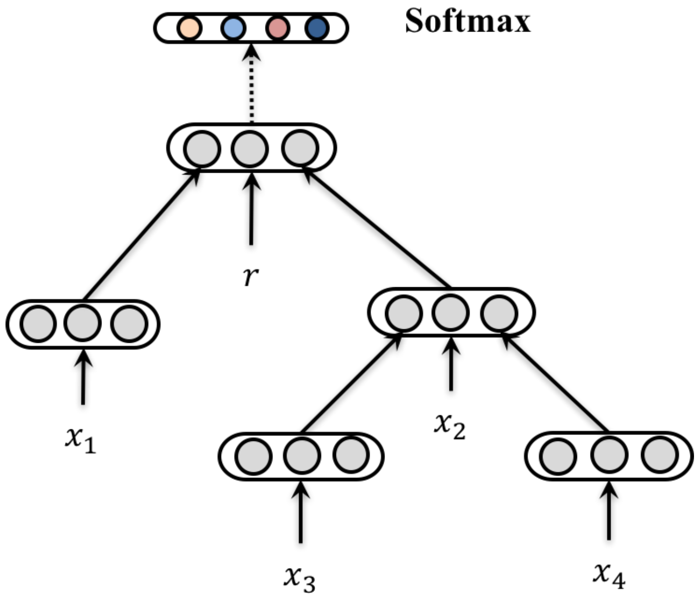
$$r_j = \sigma\left(W_r \tilde{x}_j + U_r h_{\mathcal{S}}\right)$$

$$z_j = \sigma\left(W_z \tilde{x}_j + U_z h_{\mathcal{S}}\right)$$

$$\tilde{h}_j = tanh\left(W_h \tilde{x}_j + U_h (h_{\mathcal{S}} \odot r_j)\right)$$

$$h_j = (1 - z_j) \odot h_{\mathcal{S}} + z_j \odot \tilde{h}_j$$





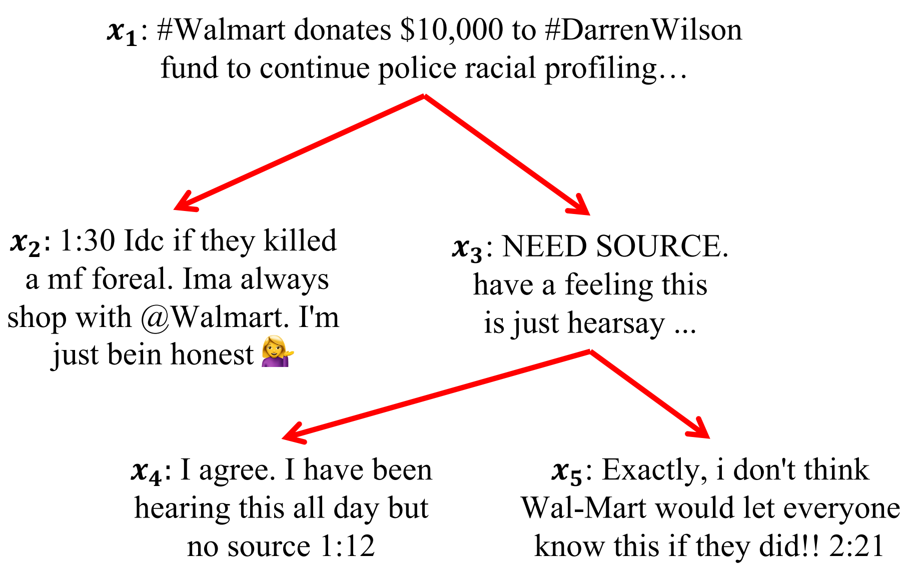
Top-down RvNN

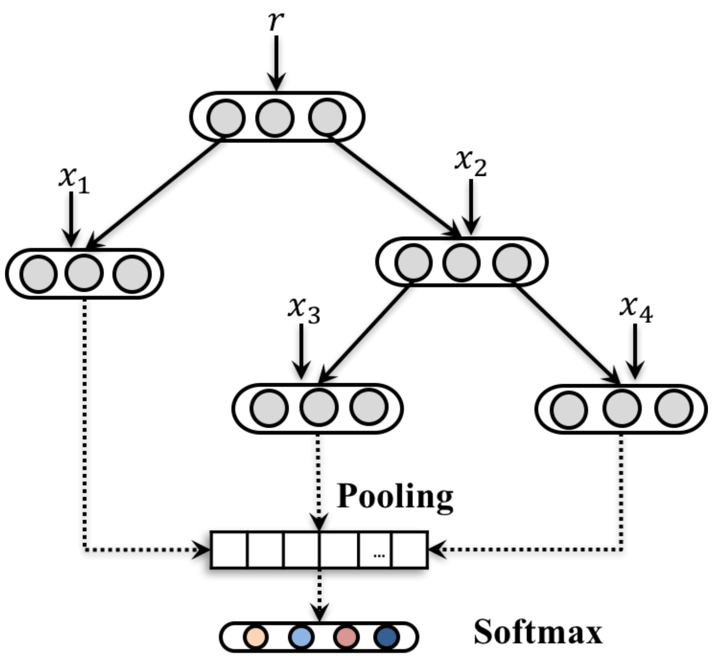
- **Input:** top-down tree
- **Structure**: recursively visit from the root node to its children until reaching all leaf nodes. (reverse Bottom-up RvNN)
- **Intuition**: rumor-indicative features are aggregated along the propagation path (e.g., if a post agree with its parent's stance, the parent's stance should be reinforced) (models how information flows from source post to the current node)

- GRU transition equation at node j

Own input Parent node

$$\tilde{x}_j = x_j E$$
$$r_j = \sigma\left(W_r \tilde{x}_j + U_r h_{\mathcal{P}(j)}\right)$$
$$z_j = \sigma\left(W_z \tilde{x}_j + U_z h_{\mathcal{P}(j)}\right)$$
$$\tilde{h}_j = tanh\left(W_h \tilde{x}_j + U_h (h_{\mathcal{P}(j)} \odot r_j)\right)$$
$$h_j = (1 - z_j) \odot h_{\mathcal{P}(j)} + z_j \odot \tilde{h}_j$$





Model Training

- Comparison:
  both of the two RvNN models aim to capture the structural properties by recursively visiting all nodes
  Bottom-up RvNN: the state of root node (i.e., source tweet) can be regard as the representation of the whole tree (can be used for supervised classification).
  Top-down RvNN: the representation of each path are eventually embedded into the hidden vector of all the leaf nodes.

- Output Layer
  learned vector of root node
  Bottom-up RvNN: $y = Softmax(Vh_0 + b)$
  Top-down RvNN: $y = Softmax(Vh_\infty + b)$
  the pooling vector over all leaf nodes

- Objective Function: $L = \sum_{n=1}^{N} \sum_{c=1}^{C} (y_c - \hat{y}_c)^2 + \lambda \|\Theta\|_2^2$
  prediction   Ground truth

- Training Procedure
  parameters are updated using efficient back-propagation through structure (Goller and Kuchler, 1996; Socher et al., 2013)

Outline

- Introduction
- Related Work
- Problem Statement
- RvNN-based Rumor Detection
- Evaluation
- Conclusion and Future Work

Data Collection

- Use two reference Tree datasets:

Statistic	Twitter15	Twitter16
# of users	276,663	173,487
# of source tweets	1,490	818
# of threads	331,612	204,820
# of non-rumors	374	205
# of false rumors	370	205
# of true rumors	372	205
# of unverified rumors	374	203
Avg. time length / tree	1,337 Hours	848 Hours
Avg. # of posts / tree	223	251
Max # of posts / tree	1,768	2,765
Min # of posts / tree	55	81

URL of the datasets:
https://www.dropbox.com/s/0jhsfwep3ywvpca/rumdetect2017.zip?dl=0

Approaches to compare with

- **DTR**: Decision tree-based ranking model using enquiry phrases to identify trending rumors (Zhao et al., 2015)
- **DTC**: Twitter information credibility model using Decision Tree Classifier (Castillo et al., 2011);
- **RFC**: Random Forest Classifier using three parameters to fit the temporal tweets volume curve (Kwon et al., 2013)
- **SVM-TS**: Linear SVM classifier using time-series structures to model the variation of social context features. (Ma et al., 2015)
- **SVM-BOW**: linear SVM classifier using bag-of-words.
- **SVM-TK** and **SVM-HK**: SVM classifier uses a Tree Kernel (Ma et al., 2017) and that uses a Hybrid Kernel (Wu et al., 2015), both model propagation structures with kernels.
- **GRU-RNN**: The RNN-based rumor detection model. (Ma et al., 2016)
- Ours (**BU-RvNN** and **TD-RvNN**): Our bottom-up and top-down recursive models.

Results on Twitter15

NR: Non-Rumor; FR: False Rumor;
TR: True Rumor; UR: Unverified Rumor;

Method	Accu.	NR	FR	TR	UR
		F1	F1	F1	F1
DTR	*0.409*	*0.501*	*0.311*	*0.364*	*0.473*
DTC	*0.454*	*0.733*	*0.355*	*0.317*	*0.415*
RFC	*0.565*	***0.810***	*0.422*	*0.401*	*0.543*
SVM-TS	*0.544*	*0.796*	*0.472*	*0.404*	*0.483*
SVM-BOW	*0.548*	*0.564*	*0.524*	*0.582*	*0.512*
SVM-HK	*0.493*	*0.650*	*0.439*	*0.342*	*0.336*
SVM-TK	*0.667*	*0.619*	*0.669*	*0.772*	*0.645*
GRU-RNN	*0.641*	*0.684*	*0.634*	*0.688*	*0.571*
BU-RvNN	*0.708*	*0.695*	*0.728*	*0.759*	*0.653*
TD-RvNN	***0.723***	*0.682*	***0.758***	***0.821***	***0.654***

hand-crafted features (e.g., user info → NR vs others)

Structural info

Linear chain input

More info loss

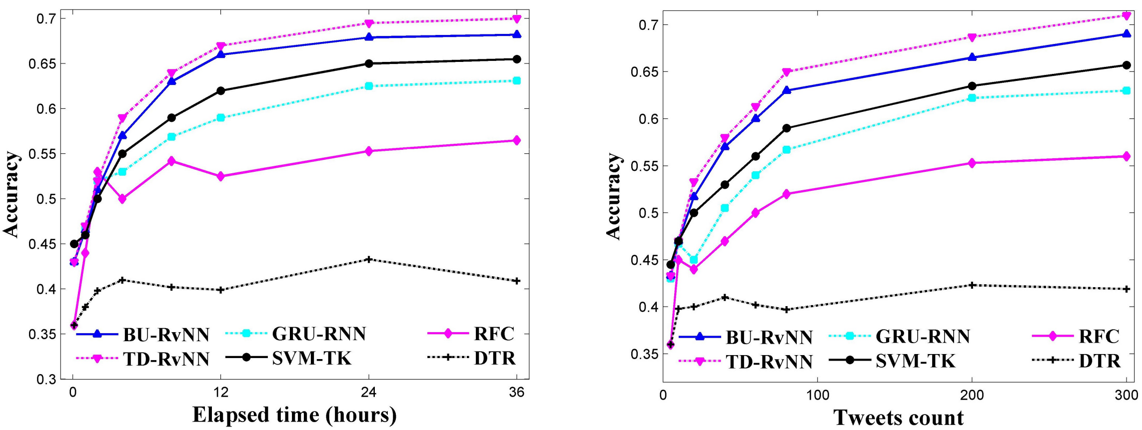
Results on Twitter16

NR: Non-Rumor; FR: False Rumor;
TR: True Rumor; UR: Unverified Rumor;

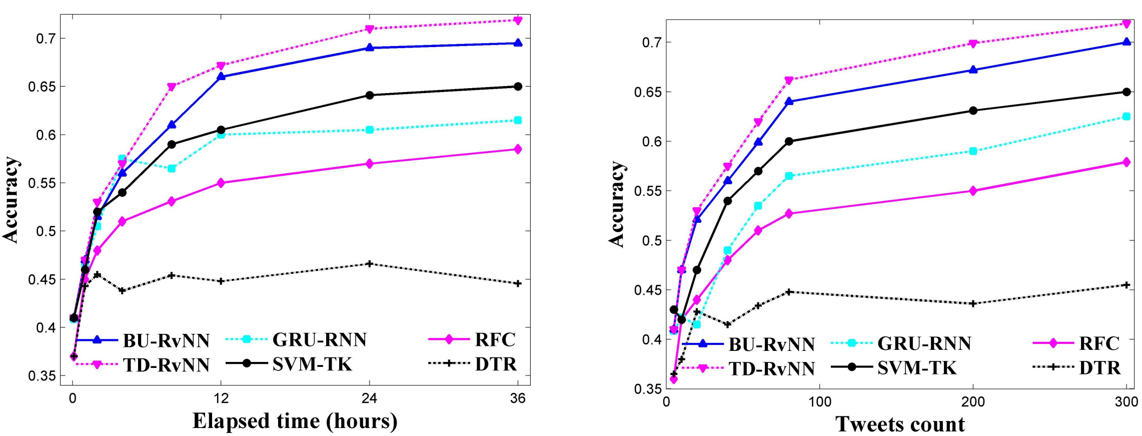
Method	Accu.	NR	FR	TR	UR
		F1	F1	F1	F1
DTR	*0.414*	*0.394*	*0.273*	*0.630*	*0.344*
DTC	*0.465*	*0.643*	*0.393*	*0.419*	*0.403*
RFC	*0.585*	*0.752*	*0.415*	*0.547*	*0.563*
SVM-TS	*0.574*	*0.755*	*0.420*	*0.571*	*0.526*
SVM-BOW	*0.585*	*0.553*	*0.556*	*0.655*	*0.578*
SVM-HK	*0.511*	*0.648*	*0.434*	*0.473*	*0.451*
SVM-TK	*0.662*	*0.643*	*0.623*	*0.783*	*0.655*
GRU-RNN	*0.633*	*0.617*	*0.715*	*0.577*	*0.527*
BU-RvNN	*0.718*	*0.723*	*0.712*	*0.779*	*0.659*
TD-RvNN	*0.737*	*0.662*	*0.743*	*0.835*	*0.708*

models without hand-crafted features

Results on Early Detection



(a) Twitter15 DATASET

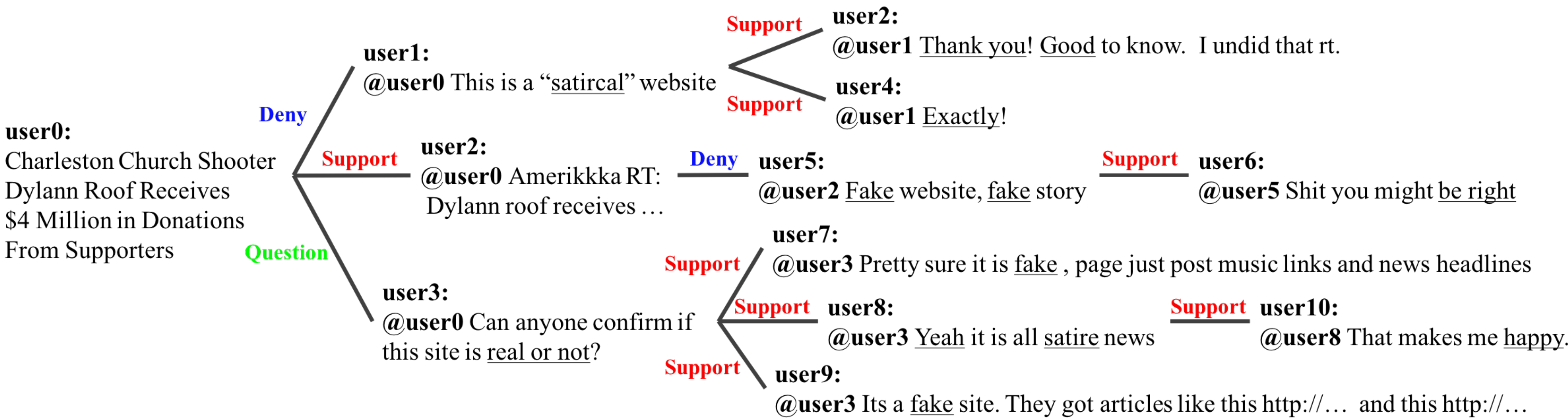


(b) Twitter16 DATASET

- In the first few hours, the accuracy of the RvNN-based methods climbs more rapidly and stabilize more quickly
- TD-RvNN and BU-RvNN only need around 8 hours or about 90 tweets to achieve the comparable performance of the best baseline model.

Early Detection Example

Example subtree of a rumor captured by the algorithm at early stage of propagation



- *Bottom-up RvNN*: a set of responses supporting the parent posts that deny or question the source post.
- *Top-down RvNN*: some patterns of propagation from the root to leaf nodes like "support→deny→support"
- *Baselines*: sequential models may be confused because the supportive key terms such as "be right", "yeah", "exactly!" dominate the responses, and the SVM-TK may miss similar subtrees by just comparing the surface words.

Outline

- Introduction
- Related Work
- Problem Statement
- RvNN-based Rumor Detection
- Evaluation
- Conclusion and Future Work

Conclusion and future work

- Propose a bottom-up and a top-down tree-structured model based on recursive neural networks for rumor detection on Twitter.
- Using propagation tree to guide the learning of representations from tweets content, such as embedding various indicative signals hidden in the structure, for better identifying rumors.
- Results on two public Twitter datasets show that our method improves rumor detection performance in very large margins as compared to state-of-the-art baselines.

- Future work:
  - Integrate other types of information such as user properties into the structured neural models to further enhance representation learning
  - Develop unsupervised models due to massive unlabeled data from social media.

Q & A