Rumor Detection on Twitter with Tree-structured Recursive Neural Networks

Jing Ma¹, Wei Gao², Kam-Fai Wong¹,³
¹The Chinese University of Hong Kong
²Victoria University of Wellington, New Zealand
³MoE Key Laboratory of High Confidence Software Technologies, China

July 15-20, 2018 – ACL 2018 @ Melbourne, 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**

[Image of a tweet by @erictucker: Anti-Trump protestors in Austin today are not as organic as they seem. Here are the busses they came in. #fakeprotests #trump2016 #austin]

[Image of a fake news article: Mark Zuckerberg is not giving $4.5 million to Facebook users who share a “thank you” message.]

[Image of a fake news article: FAKE NEWS]
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
Why such model do better?

Local characteristic:

- A reply usually respond to its **immediate** ancestor rather than the **root tweet**.
- 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 cue terms: Zhao et al. (2015)
  - Tree-kernel-based model:
    - Ma et al. (2017), Wu et al. (2015)

- 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

Without hand-crafted features
Outline

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

- Given a set of microblog posts \( R = \{r\} \), model each source tweet as a tree structure \( T(r) = < V, E > \), 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
Viva La Revolución @70torinoman - 18 Oct 2014
Walmart donates $10,000 to support Darren Wilson and the on going racist police murders #Ferguson #BoycottWalmart

Annine Mae @anniemae1000 - 19 Oct 2014
Replying to @70torinoman
@meinooooe @70torinoman I doubt they did but if it turns out to be true then Good For Them!

Melanie B @meinooooe - 19 Oct 2014
@anniemae1000 @70torinoman it'd be really inhumane if they did. If they did, They support murder basically.

Annie Mae @anniemae1000 - 19 Oct 2014
@meinooooe @70torinoman I think they support protecting their store from looters.

Melanie B @meinooooe - 19 Oct 2014
@anniemae1000 @70torinoman doubt it. They've already fixed the store & have it protected. Corporate had to have sent that.

Annie Mae @anniemae1000 - 19 Oct 2014
@meinooooe @70torinoman whatever

Viva La Revolución @70torinoman - 19 Oct 2014
@anniemae1000 @meinooooe the privilege of "whatever"

Jing Ma (CUHK)
Outline

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

\[
\tilde{x}_j = x_j E \\
h_S = \sum_{s \in \mathcal{S}(j)} h_s \\
r_j = \sigma (W_r \tilde{x}_j + U_r h_S) \\
z_j = \sigma (W_z \tilde{x}_j + U_z h_S) \\
\tilde{h}_j = \tanh (W_h \tilde{x}_j + U_h (h_S \odot r_j)) \\
h_j = (1 - z_j) \odot h_S + z_j \odot \tilde{h}_j
\]

\[\text{Input:} \quad \text{Bottom-up tree (node: a post represented as a vector of words) \qquad \text{GRU equation at node } j\]
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 $j$

$$
\begin{align*}
\tilde{x}_j &= x_j E \\
r_j &= \sigma (W_r \hat{x}_j + U_r h_{P(j)}) \\
z_j &= \sigma (W_z \hat{x}_j + U_z h_{P(j)}) \\
\tilde{h}_j &= tanh (W_h \hat{x}_j + U_h (h_{P(j)} \odot r_j)) \\
h_j &= (1 - z_j) \odot h_{P(j)} + z_j \odot \tilde{h}_j
\end{align*}
$$

$x_1$: #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
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
  Bottom-up RvNN: \( y = \text{Softmax}(Vh_0 + b) \)
  Top-down RvNN: \( y = \text{Softmax}(Vh_\infty + b) \)

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

- 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:

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

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

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

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

- hand-crafted features (e.g., user info → NR vs others)
- Structural info
- Linear chain input
- More info loss
NR: Non-Rumor; FR: False Rumor; TR: True Rumor; UR: Unverified Rumor;

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

models without hand-crafted features
Results on Early Detection

- 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.

(a) Twitter15 DATASET

(b) Twitter16 DATASET
**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.
Thank You!