Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning

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 31- August 2, 2017 – ACL 2017 @ Vancouver, Canada

*Work done when Wei Gao was in QCRI

Outline

- Introduction
- Related Work
- Tweets Propagation
- Kernel Modeling
- Evaluation
- Conclusion and Future Work

Introduction

erictucker
@erictucker

FALSE

Anti-Trump protestors in Austin today are not as organic as they seem. Here are the busses they came in. #fakeprotests #trump2016 #austin

RETWEETS 16,931 LIKES 14,521

8:43 PM - 9 Nov 2016

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

Eric Tucker, a 35-year-old co-founder of a marketing company in Austin, Tex., had just about 40 Twitter followers. But his recent tweet about paid protesters being bused to demonstrations against President-elect Donald J. Trump fueled a nationwide conspiracy theory — one that Mr. Trump joined in promoting.

Mr. Tucker's post was shared at least 16,000 times on Twitter and more than 350,000 times on Facebook. The problem is that Mr. Tucker got it wrong. There were no such buses packed with paid protesters.

But that didn't matter.

While some fake news is produced purposefully by teenagers in the Balkans or entrepreneurs in the United States seeking to make money from advertising, false information can also arise from misinformed social media posts by regular people that are seized on and spread through a hyperpartisan blogosphere.

Here, The New York Times deconstructs how Mr. Tucker's now-deleted declaration on Twitter the night after the election turned into a fake-news phenomenon. It is an example of how, in an ever-connected world where speed often takes precedence over truth, an observation by a private citizen can quickly become a talking point, even as it is being proved false.

Donald J. Trump
@realDonaldTrump
Follow

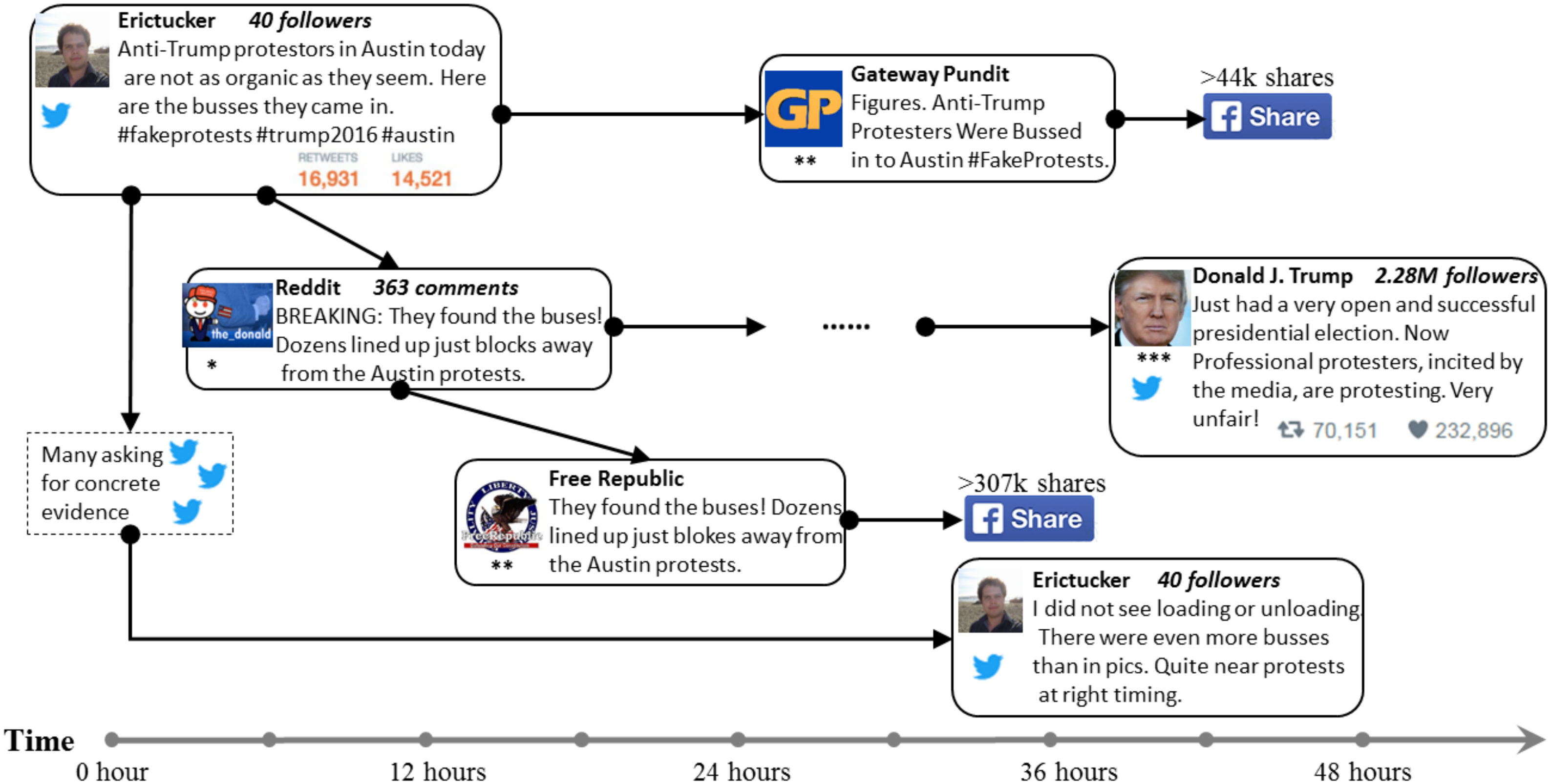
Just had a very open and successful presidential election. Now professional protesters, incited by the media, are protesting. Very unfair!

10:19 AM - 11 Nov 2016

68,261 228,412

Introduction

The fake news went viral



'*' indicates the level of influence.

Start from a grass-roots users, promoted by some influential accounts, widely spread

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.

Contributions

- Represent information spread on Twitter with propagation tree, formed by harvesting user's interactions, to capture high-order propagation patterns of rumors.
- Propose a kernel-based data-driven method to generate relevant features automatically for estimating the similarity between two propagation tees.
- Enhance the proposed model by considering propagation paths from source tweet to subtrees to capture the context of transmission.
- Release two real-world twitter datasets with finer-grained ground truth labels.

Outline

- Introduction
- Related Work
- Tweets Propagation
- Kernel Modeling
- 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)
- Kernel-based works
  - Tree kernel: syntactic parsing (Collins and Duffy, 2001)
  - Question-answering (Moschitti, 2006)
  - Semantic analysis (Moschitti, 2004)
  - Relation extraction (Zhang et al., 2008)
  - Machine translation (Sun et al., 2010)

Outline

- Introduction
- Related Work
- Tweets Propagation
- Kernel Modeling
- 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 $v = (u_v, c_v, t_v)$ provide the creator of the post, the text content and post time. And E is directed edges corresponding to response relation.

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

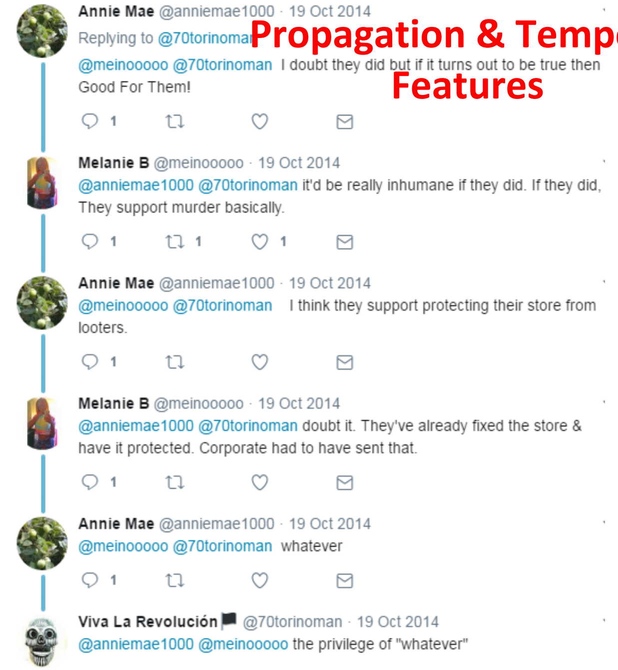
  false rumor, true rumor, non-rumor, unverified rumor

- Task 2 – detect rumor as early as possible

Propagation Structure

Text Features

User Features



Propagation & Temporal Features

Propagation tree

O: Walmart donates $10,000 to support Darren Wilson and the on going racist police murders

U1: You don't honestly believe that, do you?

U2: i honestly do

U3: …Sam Walton gave 300k to Obama's campaign? THINK.

U4: Sam Walton was dead before #Obama was born. He have wired campaign donation from heavens.

U5: where is the credible link?

……

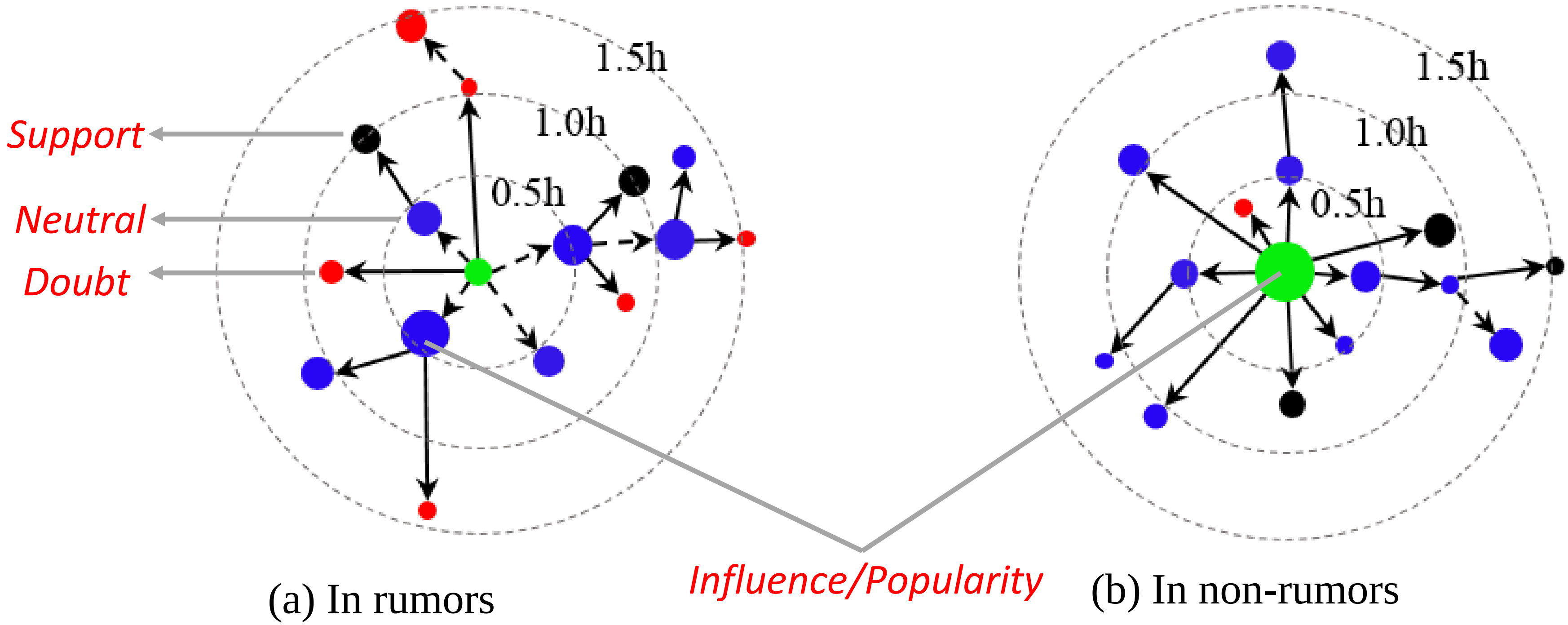
U6: Need proof of this-can't find any...

U7: not sure....sorry I see a meme trending but no proof...perhaps if we had real journalists?

U8: I'm pretty good at research-I think this is not true-plenty of other reasons to boycott WalMart. :)

……

Observation & Hypothesis



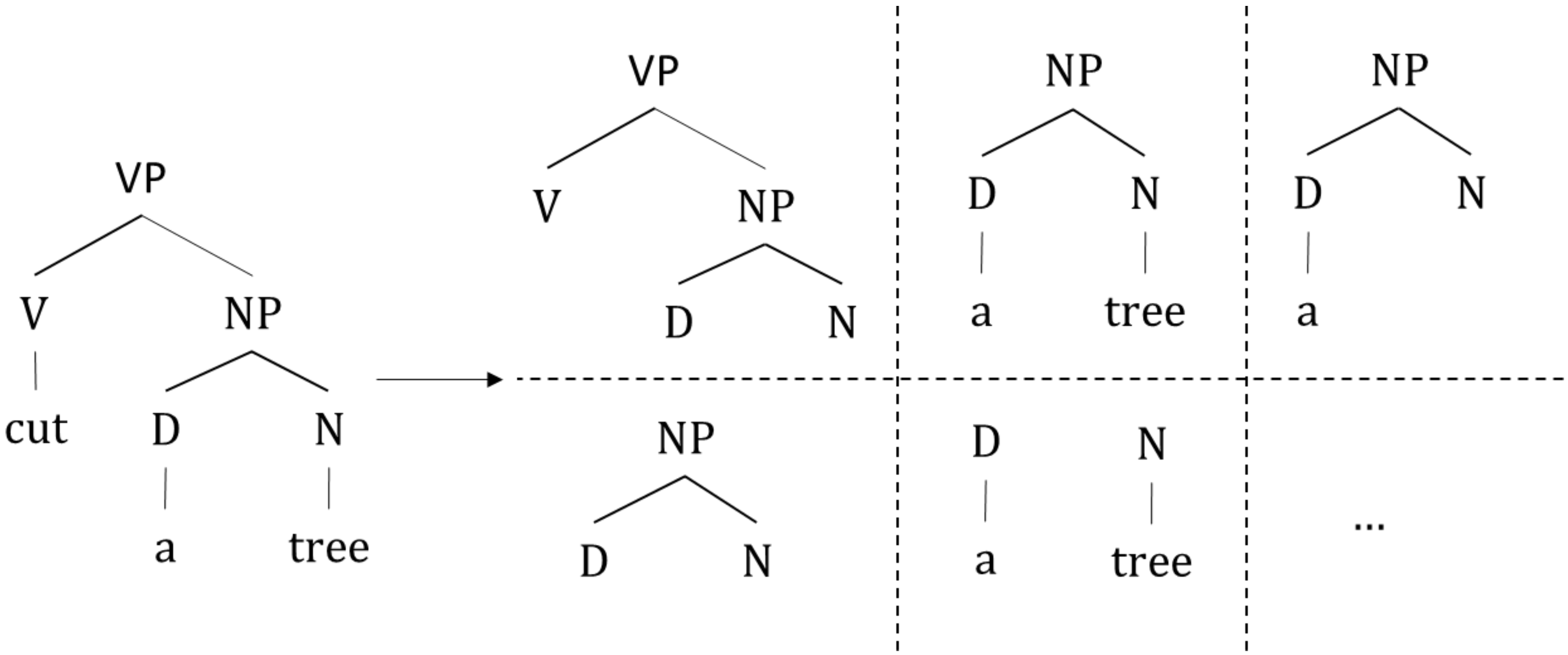
(a) In rumors (b) In non-rumors

- Content-based signals (e.g., stance) *(Zhao et al., 2015)*
- Network-based signals (e.g., relative influence) and temporal traits *(Kwon et al., 2017)*
- Our hypothesis: high-order patterns needs to/could be captured using kernel method

Outline

- Introduction
- Related Work
- Tweets Propagation
- Kernel Modeling
- Evaluation
- Conclusion and Future Work

Traditional Tree Kernel (TK)



- TK compute the syntactic similarity between two sentences by counting the common subtrees
- Kernel Function: $\sum_{v_i \in V_1} \sum_{v_j \in V_2} \Delta(v_i, v_j)$
  - $\Delta(v_i, v_j)$: common subtrees rooted at v_i and v_j

Propagation Tree Kernel (PTK)

- Why PTK?
  - Existing tree kernel cannot apply here, since in our case (1) node is a vector of continuous numerical values; (2) similarity needs to be *softly* defined between two trees instead of hardly counting on identical nodes

- Similarity Definition
  - User Similarity: $\mathcal{E}(u_i, u_j) = \|u_i - u_j\|$
  - Content Similarity: $\mathcal{J}(c_i, c_j) = \frac{|Ngram(c_i) \cap Ngram(c_j)|}{|Ngram(c_i) \cup Ngram(c_j)|}$
  - Node Similarity:

$$f\left(v_i, v_j\right) = e^{-t}\left(\alpha\mathcal{E}(u_i, u_j) + (1-\alpha)\mathcal{J}(c_i, c_j)\right)$$

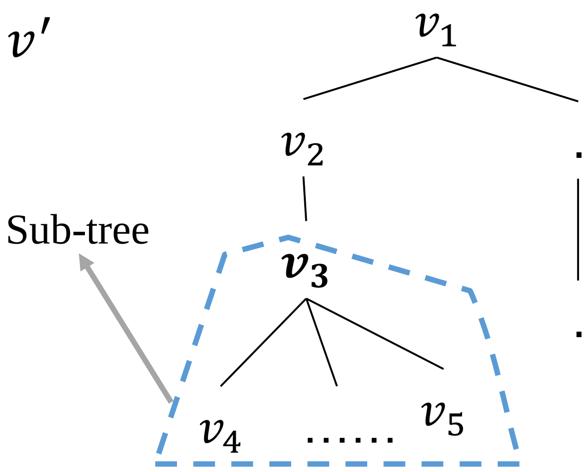
Propagation Tree Kernel

- Given two trees $T_1 =< V_1, E_1 >$ and $T_2 =< V_2, E_2 >$, PTK compute similarity between them by enumerating all similar subtrees.

- Kernel Function: $\sum_{v_i \in V_1} \Delta(v_i, v_i') + \sum_{v_j \in V_2} \Delta(v_j', v_j)$
  - v_i and v_i' are similar node pairs from V_1 and V_2 respectively

$$v_i' = \arg\max_{v_j \in V_2} f(v_i, v_j)$$

  - $\Delta(v, v')$: similarity of two subtrees rooted at v and v'



- Kernel algorithm

1) if v or v' are leaf nodes, then

$$\Delta(v, v') = f(v, v');$$

2) else

$$\Delta(v, v') = f(v, v') \prod_{k=1}^{min(nc(v), nc(v'))} (1 + \Delta(ch(v,k), ch(v',k)));$$

Context-Sensitive Extension of PTK

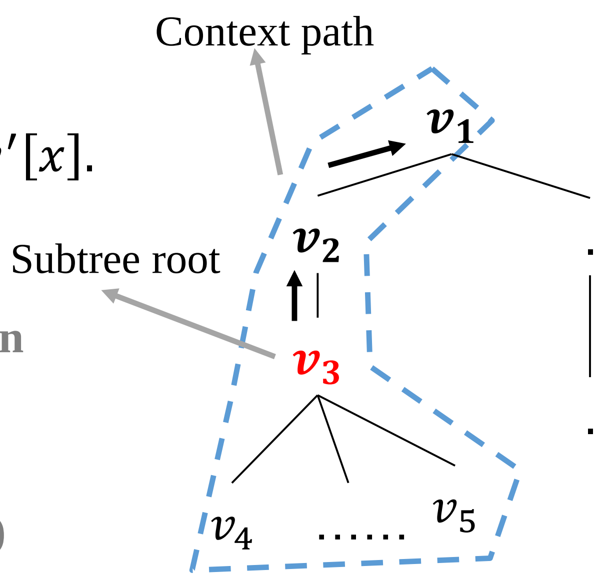
- Consider propagation paths from root node to the subtree
- Why cPTK?
  - PTK ignores the clues outside the subtrees and the route embed how the propagation happens.
  - Similar intuition to context-sensitive tree kernel *(Zhou et al., 2007)*
- Kernel Function: $\sum_{v_i \in V_1} \sum_{x=0}^{L_{v_i}^{r_1}-1} \Delta_x(v_i, v_i') + \sum_{v_j \in V_2} \sum_{x=0}^{L_{v_j}^{r_2}-1} \Delta_x(v_j', v_j)$
  - $\boldsymbol{L_v^r}$: the length of propagation path from root $\boldsymbol{r}$ to $\boldsymbol{v}$.
  - $\boldsymbol{v}[\boldsymbol{x}]$: the x-th ancestor of $\boldsymbol{v}$.
  - $\Delta_x(v, v')$: similarity of subtrees rooted at $v[x]$ and $v'[x]$.
- Kernel Algorithm

1) if $v[x]$ and $v'[x]$ are the x-th ancestor nodes of v and v' ,then

$$\Delta_x(v, v') = f(v[x], v'[x]);$$

2) else

$$\Delta_x(v, v') = \Delta(v, v')) \text{ ;i.e., PTK)}$$



Rumor Detection via Kernel Learning

- Incorporate the proposed tree kernel functions (i.e., PTK or cPTK) into a supervised learning framework, for which we utilize a kernel-based SVM classifier.

- Avoid feature engineering – the kernel function can explore an implicit feature space when calculating the similarity between two objects.

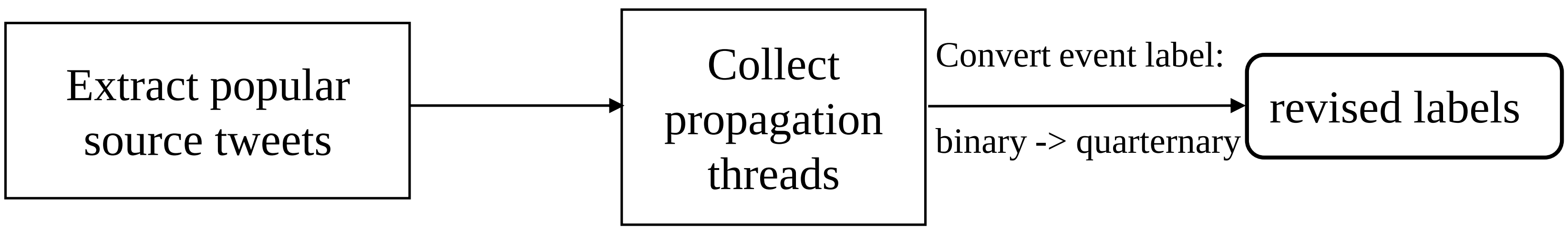
- For multi-class task, perform One vs. all, i.e., building K (# of classes) basic binary classifiers so as to separate one class from all the others.

Outline

- Introduction
- Related Work
- Tweets Propagation
- Kernel Modeling
- Evaluation
- Conclusion and Future Work

Data Collection

- Construct our propagation tree datasets based on two reference Twitter datasets:
  - Twitter15 (Liu et al, 2015)
  - Twitter16 (Ma et al, 2016)



(Source tweet: highly retweeted or replied) **(retweets: Twrench.com)**

(replies: Web crawler)

Statistics of Data Collection

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** and **SVM-RBF:** Twitter information credibility model using Decision Tree Classifier (Castillo et al., 2011); SVM-based model with RBF kernel (Yang et al., 2012)
- **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)
- **GRU:** The RNN-based rumor detection model. (Ma et al., 2016)
- **BOW:** linear SVM classifier using bag-of-words.
- **Ours (PTK and cPTK):** Our kernel based model
- **PTK- and cPTK-:** Our kernel based model with subset node features.

Results on Twitter15

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

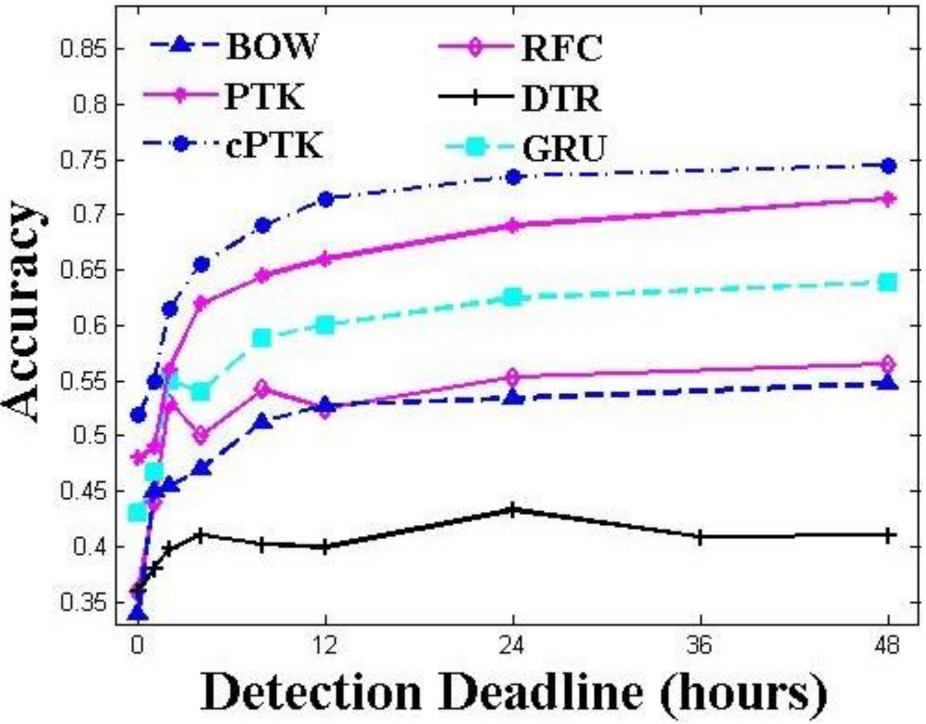
Method	Accu.	NR	FR	TR	UR
		F1	F1	F1	F1
DTR	***0.409***	***0.501***	***0.311***	***0.364***	***0.473***
SVM-RBF	***0.318***	***0.455***	***0.037***	***0.218***	***0.225***
DTC	***0.454***	***0.733***	***0.355***	***0.317***	***0.415***
SVM-TS	***0.544***	***0.796***	***0.472***	***0.404***	***0.483***
RFC	***0.565***	***0.810***	***0.422***	***0.401***	***0.543***
GRU	***0.646***	***0.792***	***0.574***	***0.608***	***0.592***
BOW	***0.548***	***0.564***	***0.524***	***0.582***	***0.512***
PTK-	***0.657***	***0.734***	***0.624***	***0.673***	***0.612***
cPTK-	***0.697***	***0.760***	***0.645***	***0.696***	***0.689***
PTK	***0.710***	***0.825***	***0.685***	***0.688***	***0.647***
cPTK	***0.750***	***0.804***	***0.698***	***0.765***	***0.733***

Results on Twitter16

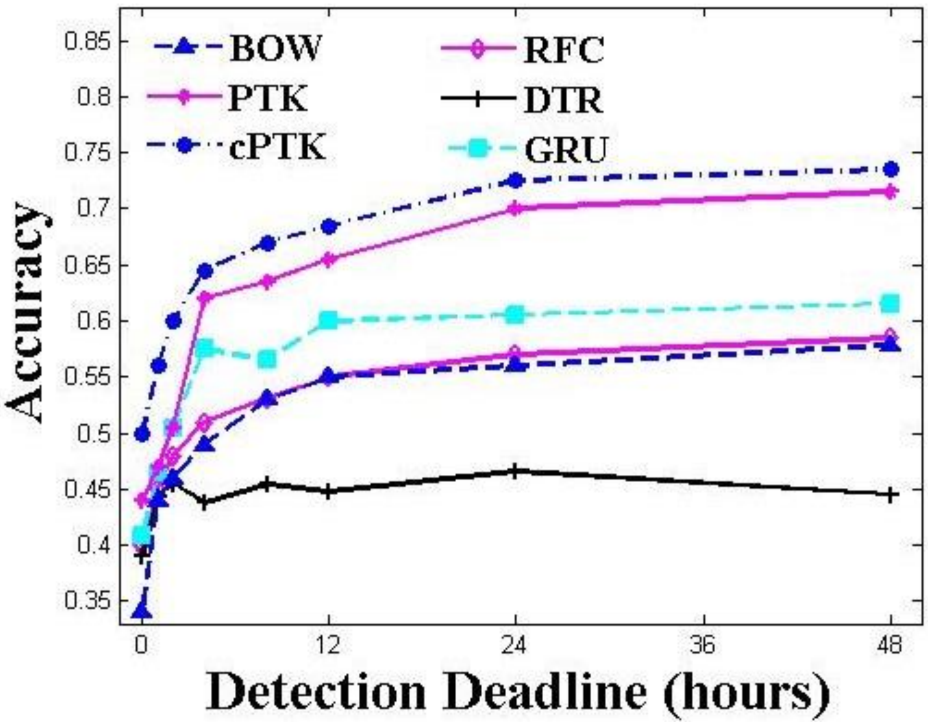
NR: Non-Rumor; FR: False Rumor;
TR: True Rumor; UR: Unverified Rumor;

Method	Accu.	NR	FR	TR	UR
		F1	F1	F1	F1
DTR	***0.414***	***0.394***	***0.273***	***0.630***	***0.344***
SVM-RBF	***0.321***	***0.423***	***0.085***	***0.419***	***0.037***
DTC	***0.465***	***0.643***	***0.393***	***0.419***	***0.403***
SVM-TS	***0.574***	***0.755***	***0.420***	***0.571***	***0.526***
RFC	***0.585***	***0.752***	***0.415***	***0.547***	***0.563***
GRU	***0.633***	***0.772***	***0.489***	***0.686***	***0.593***
BOW	***0.585***	***0.553***	***0.556***	***0.655***	***0.578***
PTK-	***0.653***	***0.673***	***0.640***	***0.722***	***0.567***
cPTK-	***0.702***	***0.711***	***0.664***	***0.816***	***0.608***
PTK	***0.722***	***0.784***	***0.690***	***0.786***	***0.644***
cPTK	***0.732***	***0.740***	***0.709***	***0.836***	***0.686***

Results on Early Detection



(a) Twitter15 DATASET

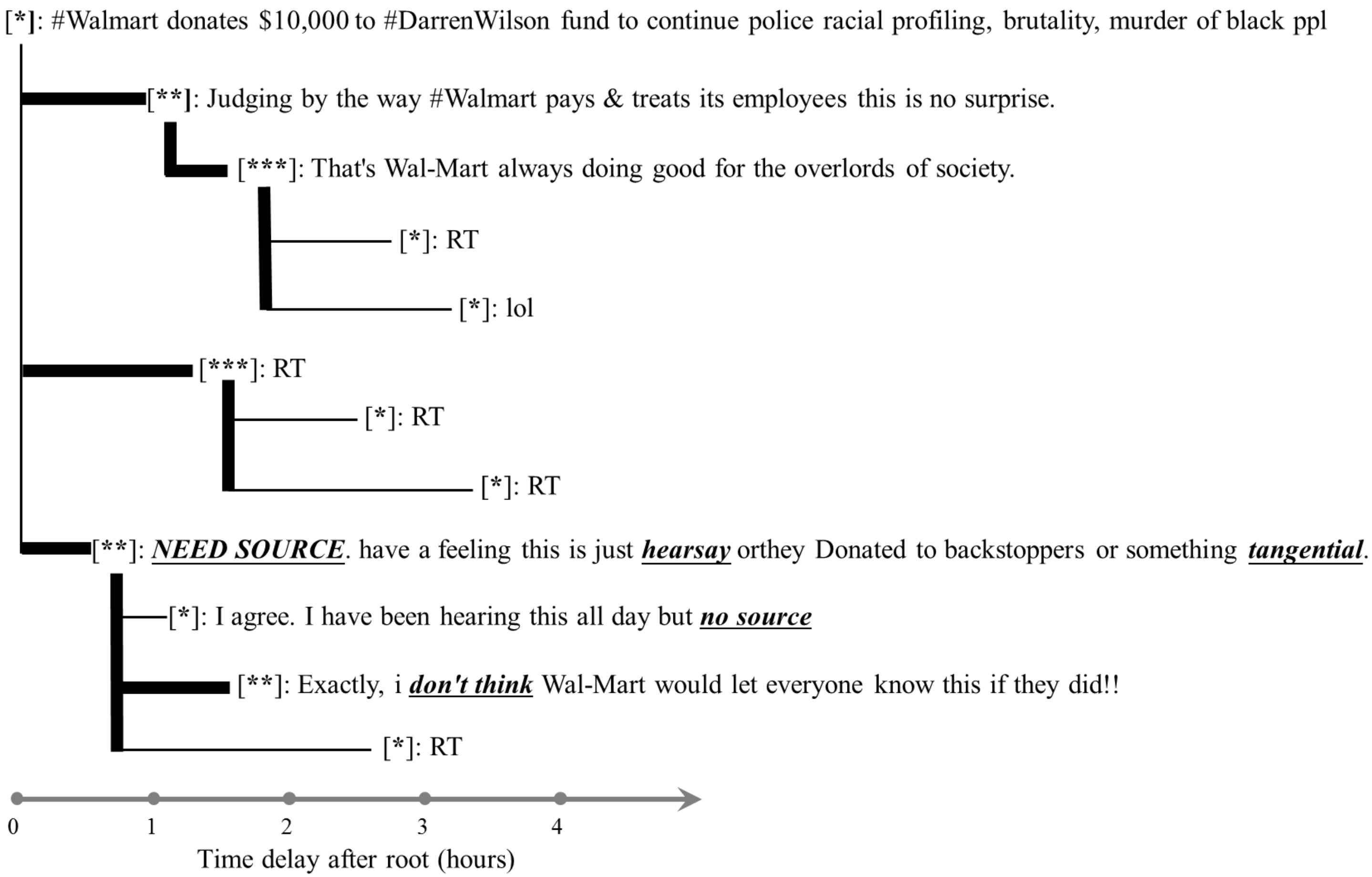


(b) Twitter16 DATASET

- In the first few hours, the accuracy of the kernel-based methods climbs more rapidly and stabilize more quickly
- cPTK can detect rumors with 72% accuracy for Twitter15 and 69.0% for Twitter16 within 12 hours, which is much earlier than the baselines and the mean official report times

Early Detection Example

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



Influential users boost its propagation, unpopular-to-popular information flow, Textual signals (underlined)

Outline

- Introduction
- Related Work
- Tweets Propagation
- Kernel Modeling
- Evaluation
- Conclusion and Future Work

Conclusion and future work

- Apply kernel learning method for rumor debunking by utilizing the propagation tree structures.
- Propagation tree encodes the spread of a source tweet with complex structured patterns and flat information regarding content, user and time associated with the tree nodes.
- Our kernel are combined under supervised framework for identifying rumors of finer-grained levels by directly measuring the similarity among propagation trees.

- Future work:
  - Explore network representation method to improve the rumor detection task.
  - Develop unsupervised models due to massive unlabeled data from social media.

Q & A