Standing on the Shoulders of Guardians: Novel Methodologies to Combat Fake News

Nguyen Vo and Kyumin Lee

Abstract Fake news and misinformation are one of the most pressing issues of modern society. In fighting against fake news, many fact-checking systems such as human-based fact-checking sites (e.g., snopes.com and politifact.com) and automatic detection systems have been developed in recent years. However, online users still keep sharing fake news even when it has been debunked. It means that early fake news detection may be insufficient and we need complementary approaches to mitigate the spread of misinformation. In this chapter, we introduce novel methods to intervene the spread of fake news and misinformation. In particular, we (1) leverage online users named *guardians*, who cite fact-checking sites as credible evidences to fact-check information in public discourse, (2) propose two novel frameworks – the first one is a recommender system to personalize fact-checking articles[1] and the second one is a text generation framework[2] to generate responses with fact-checking intention. Both frameworks are designed to increase the guardians' engagement in fact-checking activities. Experimental results showed that our recommender system improves competitive baselines significantly by 10∼20%, and the text generation framework is able to generate relevant responses and outperforms state-of-the-art models by achieving up to 30% improvement. Our qualitative study also confirms that the superiority of our generated responses compared with responses generated from the existing models.

Keywords Fact-checking · Fake news · Fact-checkers · Recommendation · Text generation

[1] https://github.com/nguyenvo09/CombatingFakeNews

[2] https://github.com/nguyenvo09/LearningFromFactCheckers

N. Vo · K. Lee (✉)
Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, USA
e-mail: nkvo@wpi.edu; kmlee@wpi.edu

K. Shu et al. (eds.), *Disinformation, Misinformation, and Fake News in Social Media*, Lecture Notes in Social Networks,
https://doi.org/10.1007/978-3-030-42699-6_10

1 Introduction

Our media landscape has been flooded by a large volume of falsified information, overstated statements, false claims, fauxtography and fake videos[3] perhaps due to the popularity, impact and rapid information dissemination of online social networks. The dramatic increase in the volume of misinformation posed severe threats to our society, degraded trustworthiness of cyberspace, and influenced the physical world. For example, $139 billion was wiped out when the Associated Press (AP)'s hacked Twitter account posted fake news regarding White House explosion with Barack Obama's injury. Owing to the detrimental impact on modern society, a large body of research work and efforts have been focused on detecting fake news and building online fact-check systems in order to debunk fake news in its early stage of dissemination.

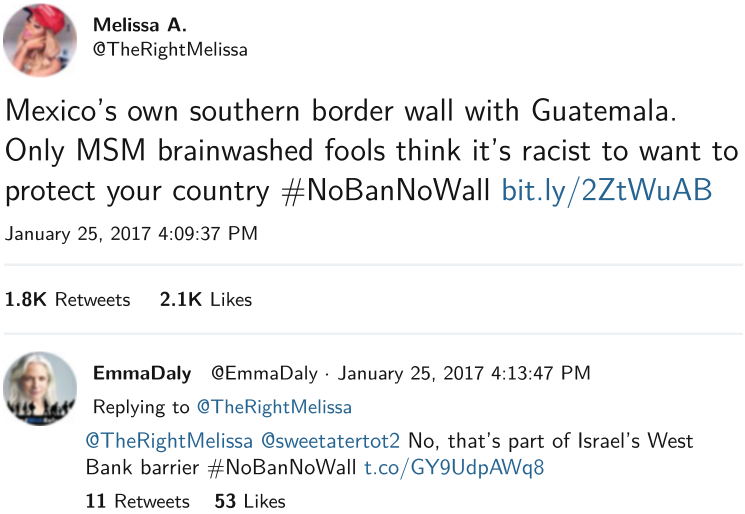
However, falsified news is still disseminated like wild fire [19, 33] despite the rise of fact-checking sites worldwide in the last half decade [11]. One possible explanation for the aforementioned phenomenon is that verifying the correctness of information may not be a common practice of the majority of people[4] since it takes time to search and read lengthy fact-checking articles. Furthermore, recent work showed that individuals tend to selectively consume news that have ideologies similar to what they believe while disregarding contradicting arguments [7, 21]. These reasons and problems indicate that using only fact-checking systems to debunk fake information is insufficient, and complementary approaches are necessary to combat fake news.

Therefore, in this chapter, we focus on online users named *guardians*, who directly engage with other users in public dialogues and convey verified information to them. Figure 1 shows a real-life conversation between two online users. The user @TheRightMelissa, called *original poster*, posts fake news about the wall between Guatemala and Mexico. After few minutes, the user @EmmaDaly refutes the misinformation by replying to the original poster and provides a fact-checking article as a supporting evidence. We call such a reply a *Direct Fact-checking tweet* (D-tweet) and the user who posts the D-tweet is called a *D-guardian*. Additionally, we notice that the D-tweet is retweeted eleven times. We call users who retweet the D-tweet secondary guardians (S-guardians) and their retweets are called secondary fact-checking tweets (S-tweets). Both *D-guardians* and *S-guardians* are called *guardians*, and both D-tweets and S-tweets are named fact-checking tweets.

In Sect. 2.1, we will show that guardians often quickly fact-checked original tweets within a day after being posted and their D-tweets could reach hundreds of millions of followers. Additionally, the likelihood to delete shares of fake news

[3] https://cnnmon.ie/2AWCCix

[4] http://go.zignallabs.com/Q1-2017-fake-news-report



Melissa A.
@TheRightMelissa

Mexico's own southern border wall with Guatemala. Only MSM brainwashed fools think it's racist to want to protect your country #NoBanNoWall bit.ly/2ZtWuAB

January 25, 2017 4:09:37 PM

1.8K Retweets **2.1K** Likes

EmmaDaly @EmmaDaly · January 25, 2017 4:13:47 PM
Replying to @TheRightMelissa
@TheRightMelissa @sweetatertot2 No, that's part of Israel's West Bank barrier #NoBanNoWall t.co/GY9UdpAWq8
11 Retweets **53** Likes

Fig. 1 A real-life fact-checking activity where the D-guardian @EmmaDaly refutes misinformation in the original tweet about the wall between Guatemala and Mexico after the original tweet was posted in few minutes

increased by 4 times when there existed a fact-checking URL in users' comments [8].

Due to the guardians' activeness and high impact on dissemination of fact-checked content, our goal is to further support them in fact-checking activities toward complementing existing fact-checking systems and combating fake news. In particular, we propose (1) a novel fact-checking URLs recommendation to recommend new and interesting fact-checking articles to guardians and (2) build a text generation framework to generate responses with fact-checking intention when original tweets are given. The fact-checking intention means either confirming or refuting content of an original tweet by providing credible evidences. Regarding two goals, this chapter shall describe these frameworks as novel methods to combat fake news.

2 Fact-Checking Article Recommendation System

In this section, we investigate who guardians are, their activeness in fact-checking activities and their impact in disseminating fact-checked contents. Based on guardians' posted fact-checking articles, we build our recommender system to personalize these articles as a way to improve guardians' engagement in fact-checking activities.

2.1 Data Collection

We employed the Hoaxy system [26] to collect a large number of D-tweets and S-tweets. In particular, we collected 231,377 unique *fact-checking tweets* from six well-known fact-checking websites – *Snopes.com*, *Politifact.com*, *FactCheck.org*, *OpenSecrets.org*, *TruthOrfiction.com* and *Hoax-slayer.net* – via the APIs provided by the Hoaxy system which internally used Twitter streaming API. The collected data consisted of 161,981 D-tweets and 69,396 S-tweets (58,821 retweets of D-tweets and 10,575 quotes of D-tweets) generated from May 16, 2016 to July 7, 2017 ($\sim$1 year and 2 month).

We removed tweets containing only base URLs (e.g., snopes.com or politifact.com) or URLs simply pointing to the background information of the websites because the tweets containing these URLs may not contain fact-checking information. After filtering, we had 225,068 fact-checking tweets consisting of 157,482 D-tweets and 67,586 S-tweets posted by 70,900 D-guardians and 45,406 S-guardians. 7,167 users played both roles of D-guardians and S-guardians. The number of unique fact-checking URLs was 7,295. In addition, we collected each guardian's recent 200 tweets. Table 1 shows the statistics of the collected dataset.

2.2 Identities of Guardians and their Activeness

As we have shown in the previous section, there were only 7,167 users (7%) who behaved as both D-guardians and S-guardians, which indicates that guardians usually focused on either fact-checking claims in conversations (i.e., being D-guardians) or simply sharing credible information (i.e., being S-guardians). Since D-guardians and S-guardians played different roles, we seek to understand which group is more enthusiastic about its role. We created two lists – a list of the number of D-tweets posted by each D-guardian and a list of the number of S-tweets posted by each S-guardian –, excluding D&S guardians who performed both roles. Then, by conducting One-sided MannWhitney U-test, we found that D-guardians were significantly more enthusiastic about their role than S-guardians (p-value$<10^{-6}$). We also found that even the D&S guardians posted relatively larger number of D-tweets than S-tweets according to Wilcoxon one-sided test (p-value$<10^{-6}$).

The majority of guardians (85.3%) posted only 1$\sim$2 fact-checking tweets. However, there were active guardians, each of whom posted over 200 fact-checking tweets. Tables 2 and 3 show the top 15 most active D-guardians and S-guardians and the number of their D-tweets and S-tweets. Red-colored *Jkj193741* and *upayr*

Table 1 Statistics of our dataset

|D-tweets|	|S-tweets|	|D-guardians|	|S-guardians|	|D&S guardians|
157,482	67,586	70,900	45,406	7,167

Table 2 Top 15 most active D-guardians and associated number of D-tweets

D-guardians and their |D-tweets|		
RandoRodeo (450)	stuartbirdman (318)	upayr (214)
pjr_cunningham (430)	ilpiese (297)	JohnOrJane (213)
TXDemocrat (384)	BreastsR4babies (255)	GreenPeaches2 (199)
Jkj193741 (355)	rankled2 (230)	spencerthayer (195)
BookRageStuff (325)	___lor__ (221)	SaintHeartwing (174)

Table 3 Top 15 most active S-guardians, and associated number of S-tweets

S-guaridans and their |S-tweets|		
Jkj193741 (294)	MrDane1982 (49)	LeChatNoire4 (35)
MudNHoney (229)	pinch0salt (46)	bjcrochet (34)
_sirtainly (75)	ActualFlatticus (42)	upayr (33)
Paul197 (66)	BeltwayPanda (36)	58isthenew40 (33)
Endoracrat (49)	EJLandwehr (36)	slasher48 (31)

Table 4 Top 15 verified guardians, and corresponding D-tweet and S-tweet count

Verified guardians and (|D-tweets| vs. |S-tweets|)		
fawfulfan (103-1)	tomcoates (37-0)	KimLaCapria (27-3)
OpenSecretsDC (37-30)	**aravosis** (29-8)	PattyArquette (29-0)
PolitiFact (41-17)	**TalibKweli** (27-8)	NickFalacci (28-0)
RobertMaguire_ (46-7)	rolandscahill (31-0)	AaronJFentress (28-0)
jackschofield (42-1)	MichaelKors (30-0)	ParkerMolloy (26-1)

guardians were especially active in joining online conversations and spreading fact-checked information.

Next, we examined whether guardians have *verified* Twitter accounts or are highly visible users, who have at least 5,000 followers. The verified accounts and highly visible users usually play an important role in social media since their fact-checking tweets can reach many audiences [13, 27]. Since the verified accounts are more trustworthy, their fact-checking tweets are often shared by many other users. In our dataset, 2,401 guardians (2.2%) had verified accounts. Table 4 shows the top 15 verified accounts. Interestingly, some of these verified accounts behaved as D&S guardians, highlighted with the blue color in the table. Particularly, @PolitiFact, and @OpenSecretsDC, the official accounts of Politifact.com and OpenSecrets.org, frequently engaged in many online conversations. 8,221 guardians (7.5%) were highly visible users. Most top verified guardians, and many top S-guardians had a large number of followers. Altogether, S-tweets of the 45,406 S-guardians reached over 200 million followers.

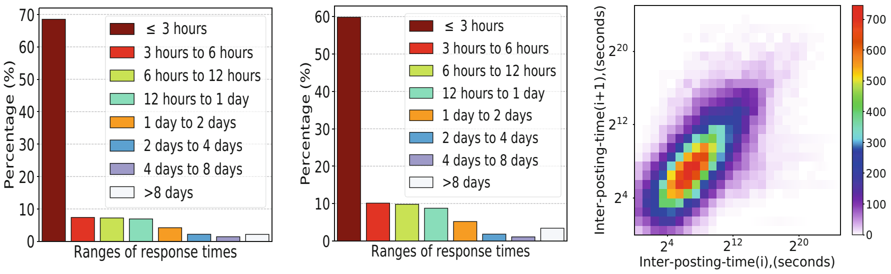
Based on the analysis, we conclude that both D-guardians and S-guardians played important roles in terms of fact-checking claims and spreading the fact-checked news to the other users. Therefore, we need both types of guardians to spread credible information.

2.3 Temporal Behavior of Guardians in Fact-Checking Activities

To further understand activeness of guardians, we examined how quickly D-guardians posted their fact-checking URLs as responses to original posters' claims in online conversations. In particular, we measured response time of a D-tweet/D-guardian as a gap between an original poster's posting time and the fact-checking D-tweet's time. We collected all response time of D-tweets, grouped them and plotted a bar chart in Fig. 2a. The mean and median of response time were 2.26 days and 34 min, respectively. 90% of D-tweets were posted within one day, indicating D-guardians quickly responded to the claims and expressed their enthusiasm by posting fact-checking URLs/tweets.

Similarly, we also measured response time of an S-tweet/S-guardian (Fig. 2b) as a gap between D-tweet's posting time and the corresponding S-tweet's posting time. The mean and median of the response time were 3.1 days and 90 min, respectively. 88.5% of S-tweets were posted within 1 day, indicating S-guardians also quickly responded and spread fact-checked information.

Finally, we measured S-guardians' inter-posting time to understand how long it took between two consecutive S-tweets, given the corresponding D-tweet. First, we grouped S-tweets based on each corresponding D-tweet, and sorted them in the ascending order of S-tweet creation time. Next, within each group, we computed inter-posting time δ_i as a gap between two consecutive S-tweets i and $i + 1$ and created pairs of inter-posting time (δ_i, δ_{i+1}). These pairs were merged across all the groups and were plotted in log2 scale in Fig. 2c. Overall, the average inter-posting time was 5 min, which means an S-tweet was posted once per 5 min by S-guardians after the corresponding D-tweet was posted. To sum up, both D-guardians and S-guardians were active and quickly responded to claims and fact-checked content.



(a) (b) (c)

Fig. 2 Ranges of response time of D-guardians and S-guardians, and inter-posting time of S-tweets. The color in (**c**) indicates the number of pairs. (**a**) D-guardians' response time. (**b**) S-guardians' response time. (**c**) S-tweets' inter-posting time

2.4 *Fact-Checking Article Recommendation Framework*

In the previous section, we found that the guardians are highly active in fact-checking activities. To encourage them to further engage in disseminating fact-checked information, we propose a recommendation model to personalize fact-checking articles. The aim of the recommendation model is to help guardians quickly access new interesting fact-checking URLs/pages so that they could embed them in their messages, correct unverified claims or misinformation, and spread fact-checked information. We use terms "fact-checking URLs", "fact-checking articles" and "URL", interchangeably.

Problem Statement Let $\mathcal{N} = \{u_1, u_2, \ldots, u_N\}$ and $\mathcal{M} = \{\ell_1, \ell_2, \ldots, \ell_M\}$ be a set of N guardians and a set of M fact-checking URLs, respectively. We view the action of embedding a fact-checking URL ℓ_j into a fact-checking tweet of guardian u_i as an interaction pair (u_i, ℓ_j). We form a matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$ where $\mathbf{X}_{ij} = 1$ if the guardian u_i posted a fact-checking URL ℓ_j. Otherwise, $\mathbf{X}_{ij} = 0$. Our main goal is to learn a model that recommends similar URLs to guardians whose interests are similar. In particular, we aim to learn matrix $\mathbf{U} \in \mathbb{R}^{N \times D}$, where each row vector $U_i^T \in \mathbb{R}^{D \times 1}$ is the latent representation of guardian u_i, and matrix $\mathbf{V} \in \mathbb{R}^{D \times M}$, where each column vector $V_j \in \mathbb{R}^{D \times 1}$ is the latent representation of URL ℓ_j. $D \ll min(M, N)$ is latent dimensions. Toward the goal, we propose our initial/basic matrix factorization model as follows:

$$\min_{\mathbf{U},\mathbf{V}} \|\Omega \odot (\mathbf{X} - \mathbf{U}\mathbf{V})\|_F^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \tag{1}$$

where $\Omega \in \mathbb{R}^{N \times M}$, and $\Omega_{ij} = 1$ if $\mathbf{X}_{ij} = 1$. Otherwise, $\Omega_{ij} = 0$. Operators $\odot$ and $\|.\|_F^2$ are Hadamard product and Frobenius norm, respectively. Finally, λ is regularization factor to avoid overfitting.

Co-ocurrence model Now, we turn to extend our basic model in Eq. 1 by further utilizing the interaction matrix $\mathbf{X}$. Inspired by [15, 20], we propose to regularize our basic model in Eq. 1 by generating two additional matrices – URL-URL co-occurrence matrix and guardian-guardian co-occurrence matrix. Our main intuition of the extension is that a pair of URLs, which were posted by the same guardian, may be similar to each other. Likewise, a pair of guardians who posted the same URLs may be alike. To better understand our proposed models, we present the word embedding model as background information.

Word embedding model Given a sequence of training words, word embedding models attempt to learn the distributed vector representation of each word. A typical example is *word2vec* proposed by Mikolov et al. [20]. Given a training word w, the main objective of the skip-gram model in *word2vec* is to predict the *context words* (i.e. the words that appear in a fixed-size context window) of w. Recently, it has been shown that training skip-gram model with negative sampling is similar to factorizing a word-context matrix named Shifted Positive Pointwise Mutual Information matrix

($SPPMI$) [14]. Given a word i and its context word j, the value $SPPMI(i, j)$ is computed as follows:

$$SPPMI(i, j) = max\{PMI(i, j) - log(s), 0\} \tag{2}$$

where $s \geq 1$ is the number of negative samples, and $PMI(i, j)$ is an element of Pointwise Mutual Information (PMI) matrix. $PMI(i, j)$ is estimated as $\log\left(\frac{\#(i,j)\times|D|}{\#(i)\times\#(j)}\right)$ where $\#(i, j)$ is the number of times that word j appears in the context window of word i. $\#(i) = \sum_j \#(i, j)$, and $\#(j) = \sum_i \#(i, j)$. $|D|$ is the total number of pairs of word and context word. Note that $PMI(i, i) = 0$ for every word i.

URL-URL co-occurrence We generate a matrix $\mathbf{R} \in \mathbb{R}^{M\times M}$ where $\mathbf{R}_{ij} = SPPMI(\ell_i, \ell_j)$ based on co-occurrence of URLs. In particular, for each URL ℓ_i posted by a specific guardian, we define its context as all other URLs ℓ_j posted by the same guardian. Based on this definition, $\#(i, j)$ means the number of guardians that posted both URL ℓ_i and ℓ_j. $\#(i, j)$ is also interpreted as the co-occurrence of URL ℓ_i and URL ℓ_j. After that, we compute $PMI(\ell_i, \ell_j)$ and $SPPMI(\ell_i, \ell_j)$ based on Eq. 2 for all pairs of ℓ_i and ℓ_j.

Guardian-Guardian co-occurrence Similarly, the context for each guardian u_i is defined as all other guardians u_j who posted the same URL with u_i. Then, $\#(i, j)$ is the number of URLs that both guardian u_i and guardian u_j commonly posted. Given this definition, we can generate a SPPMI matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$ where $\mathbf{G}_{ij} = SPPMI(u_i, u_j)$. The same value of hyper-parameter s is used for generating matrices $\mathbf{R}$ and $\mathbf{G}$.

Regularizing matrix factorization with co-occurrence matrices Our intuition is that URLs which are commonly posted by similar set of guardians are similar, and guardians who commonly posted the same set of URLs are close to each other. With that intuition, we propose loss function $\mathcal{L}_{XRG}$ – a joint matrix factorization model of three matrices $\mathbf{X}$, $\mathbf{R}$ and $\mathbf{G}$ as follows:

$$\begin{aligned}\mathcal{L}_{XRG} = &\|\Omega \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)\\ &+ \|\mathbf{R}^{mask} \odot (\mathbf{R} - \mathbf{V}^T\mathbf{K})\|_F^2 + \|\mathbf{G}^{mask} \odot (\mathbf{G} - \mathbf{UL})\|_F^2\end{aligned} \tag{3}$$

where $\mathbf{R}^{mask} \in \mathbb{R}^{M\times M}$, $\mathbf{R}^{mask}_{ij} = 1$ if $\mathbf{R}_{ij} > 0$. Otherwise, $\mathbf{R}^{mask}_{ij} = 0$. $\mathbf{G}^{mask} \in \mathbb{R}^{N\times N}$, $\mathbf{G}^{mask}_{ij} = 1$ if $\mathbf{G}_{ij} > 0$. Otherwise, $\mathbf{G}^{mask}_{ij} = 0$. Two matrices $\mathbf{K} \in \mathbb{R}^{D\times M}$ and $\mathbf{L} \in \mathbb{R}^{D\times N}$ act as additional parameters. Although our work shares similar ideas with [15], there are three key differences between our model and [15] as follows: (1) we omit bias matrices to reduce model complexity which is helpful in reducing overfitting, (2) additional matrix $\mathbf{G}$ is factorized and (3) we do not regularize parameters $\mathbf{K}$ and $\mathbf{L}$.

2.5 Integrating Auxiliary Information

In addition, we propose auxiliary information which will be integrated with Eq. 3 to improve URL recommendation performance.

Modeling social structure The social structure of guardians may reflect the homophily phenomenon indicating that guardians who follow each other may have similar interests in fact-checking URLs. To model this social structure of guardians, we first construct an unweighted undirected graph $G(V, E)$ where nodes are guardians, and an edge (u_i, u_j) between guardians u_i and u_j are formed if u_i follows u_j or u_j follows u_i. In our dataset, in total, there were 1,033,704 edges in $G(V, E)$ (density = 0.013898), which is 5.9 times higher than reported density in [31], indicating dense connections between guardians. We represent $G(V, E)$ by using an adjacency matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ where $\mathbf{S}_{ij} = 1$ if there is an edge (u_i, u_j). Otherwise, $\mathbf{S}_{ij} = 0$. Second, we use Eq. 4 as a regularization term to make latent representations of connected guardians similar to each other. Then, we formally minimize $\mathcal{L}_1$ as follows:

$$\mathcal{L}_1 = \|\mathbf{S} - \mathbf{U}\mathbf{U}^T\|_F^2 \tag{4}$$

Modeling topical interests based on recent tweets In addition to social structure, the content of recent tweets may reflect guardians' interests [1, 2, 5]. For each guardian, we build a document by aggregating his/her 200 recent tweets and then employ the Doc2Vec model [12] to learn latent representations of the document. Doc2Vec is an unsupervised learning algorithm, which automatically learns high quality representation of documents. We use Gensim[5] as implementation of the Doc2Vec, set 300 as latent dimensions of documents, and train Doc2Vec model for 100 iterations. After training Doc2Vec model, we derive cosine similarity of every pair of learned vectors to create a symmetric matrix $\mathbf{X}_{uu} \in \mathbb{R}^{N \times N}$, where $\mathbf{X}_{uu}(i, j) \in [0; 1]$ represents the similarity of document vectors of guardians u_i and u_j. Intuitively, if two guardians have similar interests, their document vectors may be similar. Thus, we regularize guardians' latent representations to make them as close as possible by minimizing the following objective function:

$$\begin{aligned}\mathcal{L}_2 &= \frac{1}{2}\sum_{i=1,j=1}^{N} \mathbf{X}_{uu}(i, j)\|U_i^T - U_j^T\|^2 \\ &= \sum_{i=1}^{N} U_i^T \mathbf{D}_{uu}(i, i) U_i - \sum_{i=1,j=1}^{N} U_i^T \mathbf{X}_{uu}(i, j) U_j \\ &= Tr(\mathbf{U}^T \mathbf{D}_{uu} \mathbf{U}) - Tr(\mathbf{U}^T \mathbf{X}_{uu} \mathbf{U}) = Tr(\mathbf{U}^T \mathcal{L}_{uu} \mathbf{U})\end{aligned} \tag{5}$$

[5] https://radimrehurek.com/gensim/

where $\mathbf{D}_{uu} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with elements on the diagonal $\mathbf{D}_{uu}(i, i) = \sum_{j=1}^{N} \mathbf{X}_{uu}(i, j)$. $Tr(.)$ is the trace of matrix, and $\mathcal{L}_{uu} = \mathbf{D}_{uu} - \mathbf{X}_{uu}$, which is a Laplacian matrix of the matrix $\mathbf{X}_{uu}$.

Modeling topical similarity of fact-checking pages We further exploit the content of fact-checking URLs (i.e., fact-checking pages) as an additional data source to improve recommendation quality. Intuitively, if the content of two URLs are similar, their latent representations should be close. Exploiting the content of a fact-checking URL has been employed in [2, 30]. In this paper, we apply a different approach, in which the Doc2Vec model is utilized to learn latent representation of URLs. Hyperparameters of the Doc2Vec model are the same as what we used for content of tweets. After training the Doc2Vec model, we derive the symmetric similarity matrix $\mathbf{X}_{\ell\ell} \in \mathbb{R}^{M \times M}$ and minimize the loss function $\mathcal{L}_3$ in Eq. 6 as a way to regulate latent representation of URLs.

Algorithm 1 GAU optimization algorithm

```
Input: Guardian-URL interaction matrix $\mathbf{X}$, URL-URL SPPMI matrix $\mathbf{R}$, Guardian-Guardian SPPMI matrix $\mathbf{G}$, social structure matrix $\mathbf{S}$, Laplacian matrix $\mathcal{L}_{uu}$ of guardians, Laplician matrix $\mathcal{L}_{\ell\ell}$ of URLs, binary matrices $\Omega$, $\mathbf{R}^{mask}$ and $\mathbf{G}^{mask}$ as indication matrices.
Output: $\mathbf{U}$ and $\mathbf{V}$
Initialize $\mathbf{U}$, $\mathbf{V}$, $\mathbf{K}$ and $\mathbf{L}$ with Gaussian distribution $\mathcal{N}(0, 0.01^2)$, $t \leftarrow 0$
while Not Converged do
    Compute $\frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{U}}$, $\frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{V}}$, $\frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{L}}$ and $\frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{K}}$ in Eq. 9
    $\mathbf{U}_{t+1} \leftarrow \mathbf{U}_t - \eta \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{U}}$
    $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t - \eta \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{V}}$
    $\mathbf{L}_{t+1} \leftarrow \mathbf{L}_t - \eta \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{L}}$
    $\mathbf{K}_{t+1} \leftarrow \mathbf{K}_t - \eta \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{K}}$
    $t \leftarrow t + 1$
return $\mathbf{U}$ and $\mathbf{V}$
```

$$\begin{aligned}
\mathcal{L}_3 &= \frac{1}{2} \sum_{i=1, j=1}^{M} \mathbf{X}_{\ell\ell}(i, j) \| V_i - V_j \|^2 \\
&= \sum_{i=1}^{M} V_i \mathbf{D}_{\ell\ell}(i, i) V_i^T - \sum_{i=1, j=1}^{M} V_i \mathbf{X}_{\ell\ell}(i, j) V_j^T \\
&= Tr(\mathbf{V}(\mathbf{D}_{\ell\ell} - \mathbf{X}_{\ell\ell})\mathbf{V}^T) \\
&= Tr(\mathbf{V}\mathcal{L}_{\ell\ell}\mathbf{V}^T)
\end{aligned} \tag{6}$$

where $\mathbf{D}_{\ell\ell} \in \mathbb{R}^{M \times M}$ is a diagonal matrix with $\mathbf{D}_{\ell\ell}(i, i) = \sum_{j=1}^{M} \mathbf{X}_{\ell\ell}(i, j)$ and $\mathcal{L}_{\ell\ell} = \mathbf{D}_{\ell\ell} - \mathbf{X}_{\ell\ell}$, which is the graph Laplacian of the matrix $\mathbf{X}_{\ell\ell}$.

2.6 Joint-Learning Fact-Checking URL Recommendation Model

Finally, we propose GAU – a joint model of **G**uardian-Guardian SPPMI matrix, **A**uxiliary information and **U**RL-URL SPPMI matrix. The objective function of our model, $\mathcal{L}_{GAU}$, is presented in Eq. 7:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{L},\mathbf{K}} \mathcal{L}_{GAU} &= \|\Omega \odot (\mathbf{X} - \mathbf{UV})\|_F^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \\ &+ \|\mathbf{R}^{mask} \odot (\mathbf{R} - \mathbf{V}^T\mathbf{K})\|_F^2 \\ &+ \|\mathbf{G}^{mask} \odot (\mathbf{G} - \mathbf{UL})\|_F^2 \\ &+ \alpha \times \|\mathbf{S} - \mathbf{UU}^T\|_F^2 \\ &+ \gamma \times Tr(\mathbf{U}^T\mathcal{L}_{uu}\mathbf{U}) \\ &+ \beta \times Tr(\mathbf{V}\mathcal{L}_{\ell\ell}\mathbf{V}^T)\end{aligned} \tag{7}$$

where $\alpha, \gamma, \beta, \lambda$ and shifted negative sampling value s are hyper parameters, tuned based on a validation set. We optimize $\mathcal{L}_{GAU}$ by using gradient descent to iteratively update parameters with fixed learning rate $\eta = 0.001$. The details of the optimization algorithm are presented in Algorithm 1. After learning **U** and **V**, we estimate the guardian u_i's preference for URL ℓ_j as: $\hat{r}_{i,j} \approx U_i V_j$. The final URLs recommended for a guardian u_i is formed based on ranking:

$$u_i : \ell_{j_1} > \ell_{j_2} > \ldots > \ell_{j_M} \rightarrow \hat{r}_{i,j_1} > \hat{r}_{i,j_2} > \ldots > \hat{r}_{i,j_M} \tag{8}$$

The derivatives of loss $\mathcal{L}_{GAU}$ with respect to parameters **U**, **V**, **K** and **L** are:

$$\begin{aligned}\frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{U}} &= -2(\Omega \odot \Omega \odot (\mathbf{X} - \mathbf{UV}))\mathbf{V}^T + 2\lambda \times (\mathbf{U}) \\ &\quad -2(\mathbf{G}^{mask} \odot \mathbf{G}^{mask} \odot (\mathbf{G} - \mathbf{UL}))\mathbf{L}^T \\ &\quad -2\alpha((\mathbf{S} - \mathbf{UU}^T + (\mathbf{S} - \mathbf{UU}^T)^T)\mathbf{U}) \\ &\quad +\gamma \times (\mathcal{L}_{uu} + \mathcal{L}_{uu}^T)\mathbf{U} \\ \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{V}} &= -2\mathbf{U}^T(\Omega \odot \Omega \odot (\mathbf{X} - \mathbf{UV})) + 2\lambda \times (\mathbf{V}) \\ &\quad -2\mathbf{K}(\mathbf{R}^{mask} \odot \mathbf{R}^{mask} \odot (\mathbf{R} - \mathbf{V}^T\mathbf{K}))^T \\ &\quad +\beta \times \mathbf{V}(\mathcal{L}_{\ell\ell} + \mathcal{L}_{\ell\ell}^T) \\ \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{L}} &= -2\mathbf{U}^T(\mathbf{G}^{mask} \odot \mathbf{G}^{mask} \odot (\mathbf{G} - \mathbf{UL})) \\ \frac{\partial \mathcal{L}_{GAU}}{\partial \mathbf{K}} &= -2\mathbf{V}(\mathbf{R}^{mask} \odot \mathbf{R}^{mask} \odot (\mathbf{R} - \mathbf{V}^T\mathbf{K}))\end{aligned} \tag{9}$$

2.7 Experimental Design and Evaluation Metrics

We were interested in selecting active and professional guardians who frequently posted fact-checking URLs since they would be more likely to spread recommended fact-checking URLs than casual guardians. We only selected guardians who used at least three distinct fact-checking URLs in their D-tweets and/or S-tweets. Altogether, 12,197 guardians were selected for training and evaluating recommendation models. They posted 4,834 distinct fact-checking URLs in total. The number of interactions was 68,684 (Sparsity:99.9%). There were 9,710 D-guardians, 6,674 S-guardians and 4,187 users who played both roles. The total number of followers of the 12,197 guardians was 55,325,364, indicating their high impact on fact-checked information propagation.

To validate our model, we randomly selected 70%, 10% and 20% URLs of each guardian for training, validation and testing. The validation data was used to tune hyper-parameters and to avoid overfitting. We repeated this evaluation scheme for five times, getting five different sets of training, validation and test data. The average results were reported. We used three standard ranking metrics such as Recall@k, MAP@k (Mean Average Precision) and NDCG@k (Normalized Discounted Cumulative Gain). We tested our model with $k \in \{5, 10, 15\}$.

2.8 Effectiveness of Auxiliary Information and SPPMI Matrices

Before comparing our GAU model with four baselines, which will be described in the following section, we first examined the effectiveness of exploiting auxiliary information and the utility of jointly factorizing SPPMI matrices. Starting from our basic model in Eq. 1, we created variants of the GAU model. Since there are many variants of GAU, we selectively report performance of the following GAU's variants:

- Our basic model (Eq. 1) (BASIC)
- BASIC + Network + URL's content (BASIC + NW + UC)
- BASIC + Network + URL's content + URL's SPPMI matrix (BASIC + NW + UC + SU)
- BASIC + URL's SPPMI matrix + Guardians' SPPMI matrix (BASIC + SU + SG)
- BASIC + Network + URL's content + SPPMI matrix of URLs + SPPMI matrix of Guardians (BASIC + NW + UC + SU + SG)
- Our GAU model

Table 5 shows performance of the variants and the GAU model. It shows the rank of each method based on the reported metrics. By adding social network information and fact-checking URL's content to Eq. 1, there was a huge climb in

Table 5 Effectiveness of using auxiliary information and co-occurrence matrices. The GAU model outperforms the other variants significantly with p-value$<$0.001

Methods	Recall@5	NDCG@5	MAP@5	Recall@10	NDCG@10	MAP@10	Recall@15	NDCG@15	MAP@15	Avg. Rank
BASIC	0.089 (6)	0.060 (6)	0.048 (6)	0.132 (6)	0.074 (6)	0.054 (6)	0.162 (6)	0.082 (6)	0.056 (6)	6.0
BASIC+NW+UC	0.099 (4)	0.068 (5)	0.055 (5)	0.148 (4)	0.084 (4)	0.062 (5)	0.183 (3)	0.093 (5)	0.064 (5)	4.4
BASIC+NW+UC+SU	0.099 (5)	0.068 (4)	0.056 (4)	0.147 (5)	0.084 (5)	0.062 (4)	0.183 (4)	0.094 (4)	0.065 (4)	4.3
BASIC+SU+SG	0.102 (3)	0.069 (3)	0.057 (3)	0.149 (3)	0.085 (3)	0.063 (3)	0.182 (5)	0.094 (3)	0.066 (3)	3.2
BASIC+NW+UC+SU+SG	0.111 (2)	0.074 (2)	0.060 (2)	0.161 (2)	0.091 (2)	0.066 (2)	0.195 (2)	0.099 (2)	0.069 (2)	2.0
Our GAU model	0.116 (1)	0.079 (1)	0.065 (1)	0.164 (1)	0.095 (1)	0.071 (1)	0.197 (1)	0.104 (1)	0.074 (1)	1.0

performance of BASIC+NW+UC over BASIC across all metrics. In particular, Recall, NDCG and MAP of BASIC+NW+UC were better than BASIC about 12.20% ± 1.31%, 13.39% ± 0.34% and 14.04% ± 0.76%, respectively (confidence interval 95%). These results confirm the effectiveness of exploiting the auxiliary information.

How about using co-occurrence SPPMI matrices of fact-checking URLs and guardians? First, when adding co-occurrence SPPMI matrix of fact-checking URL (SU) to the variant BASIC+NW+UC, we did not see much improvement across all settings. Second, when jointly factorizing two SPPMI matrices (BASIC+SU+SG) and comparing it with the variant BASIC+NW+UC, we can see that BASIC+SU+SG and BASIC+NW+UC performed equally well. Again, BASIC+SU+SG did not use any additional data sources except the interaction matrix **X**. It is an attractive benefit since it did not depend on other data sources. In other words, it reflects that regularizing the BASIC model with SPPMI matrices is comparable to adding network data and URLs' contents to the BASIC model.

So far, both auxiliary information and SPPMI matrices are beneficial to improving recommendation quality. How about combining all of them into a single model? Will performance be further improved? We turned to the variant BASIC+NW+UC+SU+SG. As expected, BASIC+NW+UC+SU+SG enhanced SU+SG by 7.90% ± 1.79% Recall, 6.58% ± 0.40% NDCG, and 5.53% ± 0.22% MAP. Its results were also higher than BASIC+NW+UC about 9.10% ± 6.15% Recall, 7.92% ± 2.50% NDCG and 7.75% ± 0.58% MAP.

Since adding auxiliary data was valuable, we now exploit another data source – 200 recent tweets' content. Consistently, adding the tweets' content indeed improved performance. The improvement of GAU over BASIC+NW+UC+SU+SG model was 4.0% Recall, 6.6% NDCG and 8.4% MAP. This improvement is statistically significant with p-value<0.001 using Wilcoxon one-sided test. Comparing the GAU with the BASIC model, we observed a dramatic increase in performance across all metrics. Specifically, Recall, NDCG and MAP were improved by 25.13% ± 10.64%, 28.64% ± 7.13% and 32% ± 4.29% respectively.

Based on the experiments, we conclude that the auxiliary data as well as co-occurrence matrices are helpful to improve recommendation quality. Adding SU+SG or NW+UC enhanced the BASIC model by 12–14%. Our GAU model performed best, improving 25∼32% compared with the BASIC model.

2.9 Performance of Our Model and Baselines

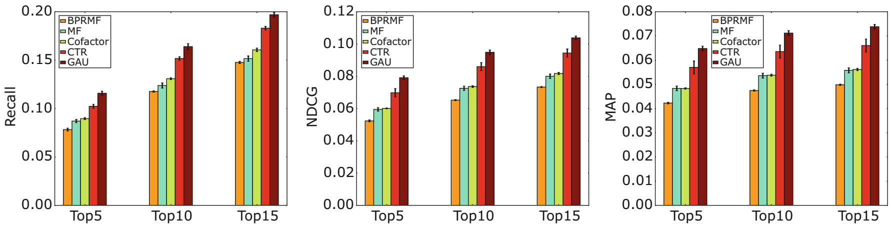
We compared our proposed model with the following four state-of-the-art collaborative filtering algorithms:

- **BPRMF** Bayesian Personalized Ranking Matrix Factorization [23] optimizes the matrix factorization model with pairwise ranking loss. It is a common baseline for item recommendation.

- **MF** Matrix Factorization (MF) [10] is a standard technique in collaborative filtering. Given an interaction matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, it factorizes $\mathbf{X}$ into two matrices $\mathbf{U} \in \mathbb{R}^{N \times D}$ and $\mathbf{V} \in \mathbb{R}^{D \times M}$, which are latent representations of users and items, respectively.
- **CoFactor** CoFactor [15] extended Weighted Matrix Factorization (WMF) by jointly decomposing interaction matrix $\mathbf{X}$ and co-occurrence SPPMI matrix for items (i.e., fact-checking URLs in this context). We set a confidence value $c_{X_{ij}=1} = 1.0$ for $X_{ij} = 1$, and we set $c_{X_{ij}=0} = 0.01$ for non-observed interaction. The number of negative samples s was grid-searched in a set $s \in \{1, 2, 5, 10, 50\}$, following the same settings as in [15].
- **CTR** Collaborative Filtering Regression [30] employed content of URLs (i.e., fact-checking pages in this context) to recommend scientific papers to users. Following exactly the best setting reported in the paper, we selected the top 8,000 words from fact-checking URLs' contents based on the mean of tf-idf values and set $\lambda_u = 0.01$, $\lambda_v = 100$, D = 200, a = 1 and b = 0.01.

To build our GAU model, we conducted the grid-search to select the best value of α, β and γ in $\{0.02, 0.04, 0.06, 0.08\}$. The number of negative samples s for constructing SPPMI matrices was in $\{1, 2, 5, 10, 50\}$. For all of the baselines and the GAU model, we set latent dimensions to $D = 100$ unless explicitly stated, and regularization value λ was grid-searched in $\{10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 7 \times 10^{-5}\}$ by default. We only report the best result of each baseline.

Figure 3 shows the performance of the four baselines and GAU. MF was better than BPRMF which was designed to optimize Area Under Curve (AUC). CTR was a very competitive baseline. This reflects the importance of fact-checking URL's content (i.e., fact-checking page) in recommending right fact-checking URLs to guardians. GAU performed better than CTR by 12.75% ± 0.95% Recall, 11.2% ± 4.6% NDCG, and 12.5% ± 2.5% MAP. GAU also outperformed CoFactor with a large margin by 25.8% ± 8.4% Recall, 29.2% ± 5.8% NDCG, and 32.6% ± 3.4% MAP (confidence interval 95%). Overall, our GAU model significantly outperformed all the baselines (p-value<0.001). The improvement over the baselines was 11~33%.

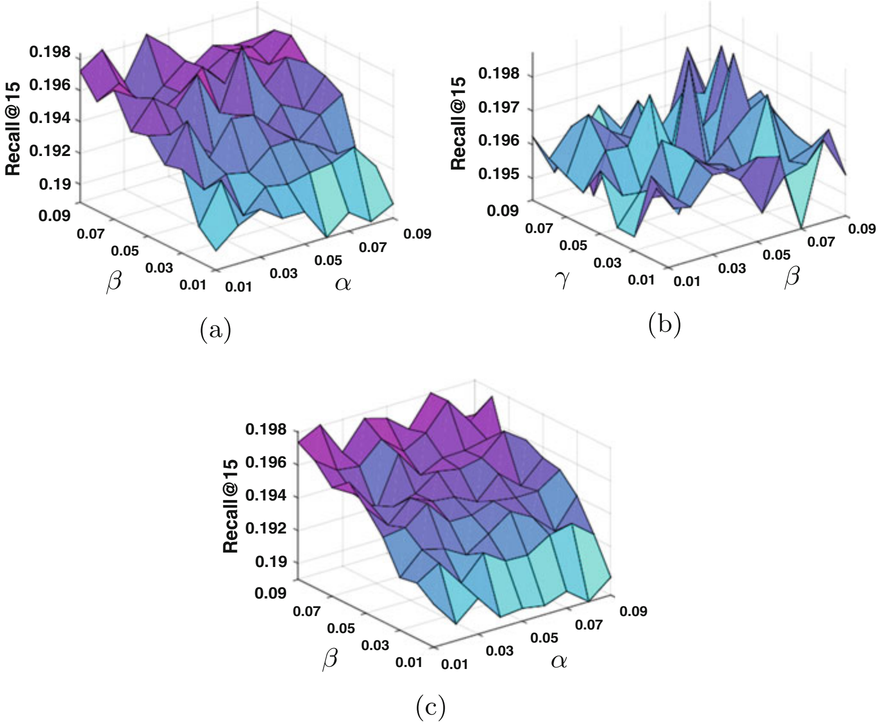


(a) Recall@k (b) NDCG@k (c) MAP@k

Fig. 3 Performance of our GAU model and 4 baselines. The GAU model outperforms the baselines (p-value < 0.001). (**a**) Recall@k. (**b**) NDCG@k. (**c**) MAP@k

2.10 Exploiting Hyper-Parameters

We investigated the impact of hyper-parameters α, β and γ on the GAU model. These hyper-parameters control the contribution of social network, fact-checking URL's content and 200 recent tweets' content to the GAU. We tested α, β and γ from 0.01 to 0.09, increasing 0.01 in each step, and then report the average recall@15, while we fixed $\lambda = 3 \times 10^{-5}$ and the number of negative samples $s = 10$. In Fig. 4a, we fixed $\beta = 0.08$ and varied α and γ. The general trend was that recall@15 gradually went up, when α and γ increased. It reached the peak, when $\alpha = 0.06$ and $\gamma = 0.06$. Next, we fixed $\alpha = 0.08$. It seems recall@15 fluctuated when varying β and γ, but the amplitude was small. The max Recall@15 was only 2.2% larger than the smallest Recall@15. Finally, γ was fixed to 0.08. The trend was similar to Fig. 4a. In general, when α, β and γ are large, the performance tends to improve, which suggests the importance of regularizing our model using the auxiliary information.



(a)

(b)

(c)

Fig. 4 Parameter sensitivity. (**a**) $\beta = 0.08$. (**b**) $\alpha = 0.08$. (**c**) $\gamma = 0.08$

2.11 Discussion

So far, we identified who guardians are and their temporal behavior. Although there are highly active guardians in fact-checking guardians, most guardians only posted 1$\sim$2 fact-checking tweets. Therefore, we only target active guardians, who posted at least 3 fact-checking URLs since guardians may continue to be active in spreading fact-checked information in the future. Another observation is that the top verified guardians seem not to be active in the covered time period. This phenomenon may be explained by the fact that these verified guardians may be cautious about what they should post to their followers. From our experiments, we showed that integrating auxiliary information is useful in improving recommendation quality. Although our model outperforms baselines, there are considerable space to improve our model. For example, we may utilize contents of original tweets, temporal factors and activeness of guardians. Deep Learning architectures may help us improve our model. We leave these directions for future exploration.

3 Fact-Checking Responses Generation Framework

In this section, we turn our attention to generating responses with fact-checking intention to help guardians fact-check information faster and as a result increase their engagement in fact-checking activities. Since S-tweets are mostly copies of Direct Fact-checking tweets (D-tweets), we focus on generating D-tweets when an original tweet is given.

3.1 Datasets of Original Tweets and Direct Fact-Checking Tweets

Since training an effective text generation framework requires large number of pairs of original tweets and D-tweets, we extend our dataset in Sect. 2.1 with additional D-tweets collected from Hoaxy system. Totally, we collected 247,436 distinct D-tweets posted between May 16, 2016 and May 26, 2018. We removed non-English D-tweets, and D-tweets containing fact-checking URLs linked to non-article pages such as the main page and about page of a fact-checking site. Then, among the remaining D-tweets, if its corresponding original tweet was deleted or was not accessible via Twitter APIs because of suspension of an original poster, we further filtered out the D-tweets. As a result, 190,158 D-tweets and 164,477 distinct original tweets were remained.

To further ensure that each of the remaining D-tweets reflected fact-checking intention and make a high quality dataset, we only kept a D-tweet whose fact-checking article was rated as true or false. Our manual verification of 100 random samples confirmed that D-tweets citing fact-checking articles with true or false label contained clearer fact-checking intention than D-tweets with other labels such as half true or mixture. In other words, D-tweets associated with mixed labels were discarded. After the pre-processing steps, our final dataset consisted of 73,203 D-tweets and 64,110 original tweets posted by 41,732 distinct D-guardians, and 44,411 distinct original posters, respectively. We use this dataset in the following sections.

3.2 Response Generation Framework

Formally, given a pair of an original tweet and a D-tweet, the original tweet x is a sequence of words $x = \{x_i | i \in [1; N]\}$ and the D-tweet is another sequence of words $y = \{y_j | j \in [1; M]\}$, where N and M are the length of the original tweet and the length of D-tweet, respectively. We inserted a special token `<s>` as a starting token into every D-tweet. Drawing inspiration from [18], we propose and build a framework as shown in Fig. 5 that consists of three main components: (i) the shared word embedding layer, (ii) the encoder to capture representation of the original tweet and (iii) the decoder to generate a D-tweet. Their details are as follows:

Shared Word Embedding Layer For every word x_i in the original tweet x, we represent it as a one-hot encoding vector $x_i \in \mathbb{R}^V$ and embed it into a D-dimensional vector $\mathbf{x}_i \in \mathbb{R}^D$ as follows: $\mathbf{x}_i = \mathbf{W}_e x_i$, where $\mathbf{W}_e \in \mathbb{R}^{D \times V}$ is an embedding matrix and V is the vocabulary size. We use the same word embedding matrix $\mathbf{W}_e$ for the D-tweet. In particular, for every word y_i (represented as one-hot vector $y_i \in \mathbb{R}^V$) in the D-tweet y, we embed it into a vector $\mathbf{y}_i = \mathbf{W}_e y_i$. The embedding matrix $\mathbf{W}_e$ is a learned parameter and could be initialized by either pre-trained word vectors (e.g.

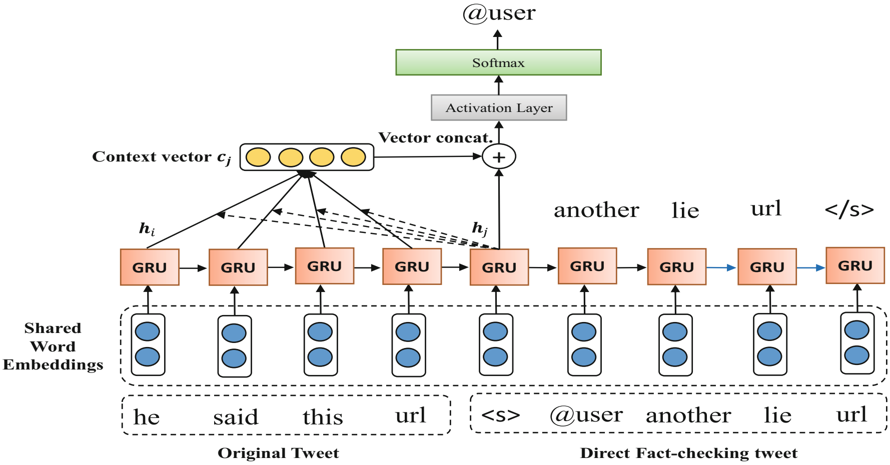


Fig. 5 Our proposed framework to generate responses with fact-checking intention

Glove vectors) or random initialization. Since our model is designed specifically for fact-checking domain, we initialized $\mathbf{W}_e$ with Normal Distribution $\mathcal{N}(0, 1)$ and trained it from scratch. By using a shared $\mathbf{W}_e$, we could reduce the number of learned parameters significantly compared with [18].

Encoder The encoder is used to learn latent representation of the original tweet x. We adopt a Recurrent Neural Network (RNN) to represent the encoder due to its large capacity to condition each word x_i on all previous words $x_{<i}$ in the original tweet x. To overcome the vanishing or exploding gradient problem of RNN, we choose Gated Recurrent Unit (GRU) [6]. Formally, we compute hidden state $\mathbf{h}_i \in \mathbb{R}^H$ at time-step ith in the encoder as follows:

$$\mathbf{h}_i = GRU(\mathbf{x}_i, \mathbf{h}_{i-1}) \tag{10}$$

where the GRU is defined by the following equations:

$$\begin{aligned}
\mathbf{z}_i &= \sigma(\mathbf{x}_i\mathbf{W}_z + \mathbf{h}_{i-1}\mathbf{U}_z)\\
\mathbf{r}_i &= \sigma(\mathbf{x}_i\mathbf{W}_r + \mathbf{h}_{i-1}\mathbf{U}_r\big)\\
\tilde{\mathbf{h}}_i &= tanh\big(\mathbf{x}_i\mathbf{W}_o + (\mathbf{r}_i \odot \mathbf{h}_{i-1})\mathbf{U}_o)\\
\mathbf{h}_i &= (1 - \mathbf{z}_i) \odot \tilde{\mathbf{h}}_i + \mathbf{z}_i \odot \mathbf{h}_{i-1}
\end{aligned} \tag{11}$$

where $\mathbf{W}_{[z,r,o]}$, $\mathbf{U}_{[z,r,o]}$ are learned parameters. $\tilde{\mathbf{h}}_i$ is the new updated hidden state, $\mathbf{z}_i$ is the update gate, $\mathbf{r}_i$ is the reset gate, $\sigma(.)$ is the sigmoid function, $\odot$ is element wise product, and $\mathbf{h}_0 = \mathbf{0}$. After going through every word of the original tweet x, we have hidden states for every time-step $\mathbf{X} = [\mathbf{h}_1 \oplus \mathbf{h}_2 \oplus \ldots \oplus \mathbf{h}_N] \in \mathbb{R}^{H\times N}$, where $\oplus$ denotes concatenation of hidden states. We use the last hidden state $\mathbf{h}_N$ as features of the original tweet $\mathbf{x} = \mathbf{h}_N$.

Decoder The decoder takes $\mathbf{x}$ as the input to start the generation of a D-tweet. We use another GRU to represent the decoder to generate a sequence of tokens $y = \{y_1, y_2, \ldots, y_M\}$. At each time-step jth, the hidden state h_j is computed by another GRU: $\mathbf{h}_j = GRU(\mathbf{y}_j, \mathbf{h}_{j-1})$ where initial hidden states are $\mathbf{h}_0 = \mathbf{x}$. To provide additional context information when generating word y_j, we apply an attention mechanism to learn a weighted interpolation context vector $\mathbf{c}_j$ dependent on all of the hidden states output from all time-steps of the encoder. We compute $\mathbf{c}_j = \mathbf{X}\mathbf{a}_j$ where each component $\mathbf{a}_{ji}$ of $\mathbf{a}_j \in \mathbb{R}^N$ is the alignment score between the jth word in the D-tweet and the ith output from the encoder. In this study, $\mathbf{a}_j$ is computed by one of the following ways:

$$\mathbf{a}_j = \begin{cases} softmax(\mathbf{X}^T\mathbf{h}_j) & \text{Dot Attention}\\ softmax(\mathbf{X}^T\mathbf{W}_a\mathbf{h}_j) & \text{Bilinear Attention}\end{cases} \tag{12}$$

where softmax(.) is a softmax activation function and $\mathbf{W}_a \in \mathbb{R}^{H \times H}$ is a learned weight matrix. Note that we tried to employ other attention mechanisms including additive attention [3] and concat attention [18] but the above attention mechanisms in Eq. 12 produced better results. After computing the context vector $\mathbf{c}_j$, we concatenate $\mathbf{h}_j^T$ with $\mathbf{c}_j^T$ to obtain a richer representation. The word at jth time-step is predicted by a softmax classifier:

$$\hat{\mathbf{y}}_j = softmax\big(\mathbf{W}_s \tanh\big(\mathbf{W}_c[\mathbf{c}_j^T \oplus \mathbf{h}_j^T]^T\big)\big) \tag{13}$$

where $\mathbf{W}_c \in \mathbb{R}^{O \times 2H}$, and $\mathbf{W}_s \in \mathbb{R}^{V \times O}$ are weight matrices of a two-layer feedforward neural network and O is the output size. $\hat{\mathbf{y}}_j \in \mathbb{R}^V$ is a probability distribution over the vocabulary. The probability of choosing word v_k in the vocabulary as output is:

$$p(y_j = v_k | y_{j-1}, y_{j-2}, \ldots, y_1, \mathbf{x}) = \hat{\mathbf{y}}_{jk} \tag{14}$$

Therefore, the overall probability of generating the D-tweet y given the original tweet x is computed as follows:

$$p(y|x) = \prod_{j=1}^{M} p(y_j | y_{j-1}, y_{j-2}, \ldots, y_1, \mathbf{x}) \tag{15}$$

Since the entire architecture is differentiable, we jointly train the whole network with Teacher Forcing via Adam optimizer by minimizing the negative conditional log-likelihood for m pairs of the original tweet $x^{(i)}$ and the D-tweet $y^{(i)}$ as follows:

$$\min_{\theta_e, \theta_d} \mathcal{L} = -\sum_{i=1}^{m} \log p(y^{(i)} | x^{(i)}; \theta_e, \theta_d) \tag{16}$$

where θ_e and θ_d are the parameters of the encoder and the decoder, respectively. At test time, we used beam search to select top K generated responses. The generation process of a D-tweet is ended when an end-of-sentence token (e.g. </s>) is emitted.

3.3 Evaluation

In this section, we thoroughly evaluate our models namely **FCRG-DT** (based on dot attention in Eq. 12) and **FCRG-BL** (based on bilinear attention in Eq. 12) quantitatively and qualitatively. Since our methods are deterministic models, we compare them with state-of-the-art baselines in this direction.

- **SeqAttB:** Shang et al. [25] proposed a hybrid model that combines global scheme and local scheme [3] to generate responses for original tweets on Sina

Weibo. This model is one of the first work that generate responses for short text conversations.

- **HRED:** It [24] employs hierarchical RNNs for capturing information in a long context. HRED is a competitive method and a commonly used baseline for dialog generation systems.
- **our FCRG-BL:** This model uses the bilinear attention.
- **our FCRG-DT:** This model uses the dot attention.

Data Processing Similar to [24] in terms of text generation, we replaced numbers with `<number>` and personal names with `<person>`. Words that appeared less than three times were replaced by `<unk>` token to further mitigate the sparsity issue. Our vocabulary size was 15,321. The min, max and mean |tokens| of the original tweets were 1, 89 and 19.1, respectively. The min, max and mean |tokens| of D-tweets were 3, 64 and 12.3, respectively. Only 791 (1.2%) original tweets contained 1 token which is mostly a URL.

Experimental Design We randomly divided 73,203 pairs of the original tweets and D-tweets into training/validation/test sets with a ratio of 80%/10%/10%, respectively. The validation set was used to tune hyperparameters and for early stopping. At test time, we used the beam search to generate 15 responses per original tweet (beam size=15), and report the average results. To select the best hyperparameters, we conducted the standard grid search to choose the best value of a hidden size $H \in \{200, 300, 400\}$, and an output size $O \in \{256, 512\}$. We set word embedding size D to 300 by default unless explicitly stated. The length of the original tweets and D-tweets were set to the maximum value $N = 89$ and $M = 64$, respectively. The dropout rate was 0.2. We used Adam optimizer with fixed learning rate $\lambda = 0.001$, batch size $b = 32$, and gradient clipping was 0.25 to avoid exploded gradient. The same settings are applied to all models for the fair comparison.

A well known problem of the RNN-based decoder is that it tends to generate short responses. In our domain, examples of commonly generated responses were *fake news url.*, *you lie url.*, and *wrong url.* Because a very short response may be less interesting and has less power to be shared, we forced the beam search to generate responses with at least τ tokens. Since 92.4% of D-tweets had |tokens| $\geq$ 5, and 60% D-tweets had |tokens| $\geq$ 10, we chose $\tau \in \{0, 5, 10\}$. In practice, fact-checkers can choose their preferred |tokens| of generated responses by varying τ.

Evaluation Metrics To measure performance of our models and baselines, we adopted several syntactic and semantic evaluation metrics used in the prior works. In particular, we used word overlap-based metrics such as BLEU scores [22], ROUGE-L [16], and METEOR [4]. These metrics evaluate the amount of overlapping words between a generated response and a ground-truth D-tweet. The higher score indicates that the generated response are close/similar to the ground-truth D-tweet syntactically. In other words, the generated response and the D-tweet have a large number of overlapping words. Additionally, we also used embedding metrics (i.e. Greedy Matching and Vector Extrema) [17]. These metrics usually estimate sentence-level vectors by using some heuristic to combine the individual word vectors in the sentence. The sentence-level vectors between a generated response and the ground-truth D-tweet are compared by a measure such as cosine similarity. The higher value means the response and the D-tweet are semantically similar.

Quantitative Results Based on Word Overlap-Based Metrics In this experiment, we quantitatively measure performances of all models by using BLEU, ROUGE-L, and METEOR. Table 6 shows results in the test set. Firstly, our FCRG-DT and FCRG-BL performed equally well, and outperformed the baselines – SeqAttB and HRED. In practice, FCRG-DT model is more preferable due to fewer parameters compared with FCRG-BL. Overall, our models outperformed SeqAttB perhaps because fusing global scheme (i.e. the last hidden state of the encoder) and output hidden state of every time-step ith in the encoder may be less effective than using only the latter one to compute context vector $\mathbf{c}_j$. HRED model utilized only global context without using context vector $\mathbf{c}_j$ in generating responses, leading to suboptimal results compared with our models.

Under no constraints on |tokens| of generated responses, our FCRG-DT achieved 6.24% ($p < 0.001$) improvement against SeqAttB on BLEU-3 according to Wilcoxon one-sided test. In BLEU-4, FCRG-DT improved SeqAttB by 7.32% and HRED by 7.76% ($p < 0.001$). In ROUGE-L, FCRG-DT improved SeqAttB and HRED by 3.32% and 4.31% with $p < 0.001$, respectively. In METEOR, our FCRG-DT and FCRG-BL achieved comparable performance with the baselines.

When |tokens| ≥ 5, we even achieve better results. The improvements of FCRG-DT over SeqAttB were 7.05% BLEU-3, 7.37% BLEU-4 and 3.25% ROUGE-L ($p < 0.001$). In comparison with HRED, the improvements of FCRG-DT were 5.25% BLEU-3, 5.64% BLEU-4, and 2.97% ROUGE-L ($p < 0.001$). Again, FCRG-DT are comparable with SeqAttB and HRED in METEOR measurement.

When |tokens| ≥ 10, there was a decreasing trend across metrics as shown in Table 6. It makes sense because generating longer response similar with a ground-truth D-tweet is much harder problem. Therefore, in reality, the Android messaging service recommends a very short reply (e.g., okay, yes, I am indeed) to reduce inaccurate risk. Despite the decreasing trend, our FCRG-DT and FCRG-BL improved the baselines by a larger margin. In particular, in BLEU-3, FCRG-DT outperformed SeqAttB and HRED by 17.9% and 16.0% ($p < 0.001$), respectively. For BLEU-4, the improvements of FCRG-DT over SeqAttB and HRED were 13.02% and 11.74% ($p < 0.001$), respectively. We observed consistent improvements over the baselines in ROUGE-L and METEOR. Overall, our models outperformed the baselines in terms of all of the word overlap-based metrics.

Quantitative Results Based on Embedding Metrics We adopted two embedding metrics to measure semantic similarity between generated responses and ground-truth D-tweets [17]. Again, we tested all the models under three settings as shown in Table 6. Our FCRG-DT performed best in all embedding metrics. Specifically, FCRG-DT outperformed SeqAttB by 3.98% and HRED by 6.00% improvements with $p < 0.001$ in Greedy Matching. FCRG-DT's improvements over SeqAttB and HRED were 26.24% and 5.62% ($p < 0.001$), respectively in Vector Extrema. When |tokens| ≥ 5, our FCRG-DT also outperformed the baselines in both Greedy

Table 6 Performance of our models and baselines. Our models outperformed baselines with p-value < 0.001

τ	Model	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	Greedy Mat.	Vector Ext.	Avg. Rank
$\tau = 0$	SeqAttB	7.15 (4)	4.05 (4)	3.26 (3)	26.47 (3)	17.66 (3)	43.57 (3)	15.84 (4)	3.43
	HRED	7.30 (3)	4.07 (3)	3.25 (4)	26.22 (4)	17.55 (4)	42.73 (4)	18.93 (3)	3.57
	FCRG-BL	**7.68** (1)	4.27 (2)	3.41 (2)	27.14 (2)	**17.87** (1)	43.71 (2)	**20.24** (1)	1.57
	FCRG-DT	7.64 (2)	**4.30** (1)	**3.50** (1)	**27.35** (1)	17.75 (2)	**45.30** (1)	19.99 (2)	1.43
$\tau = 5$	SeqAttB	7.47 (4)	4.09 (4)	3.18 (4)	26.17 (4)	17.72 (3)	41.04 (4)	14.69 (4)	3.86
	HRED	7.63 (3)	4.16 (3)	3.23 (3)	26.24 (3)	17.62 (4)	41.93 (3)	18.85 (3)	3.14
	FCRG-BL	7.93 (2)	4.29 (2)	3.30 (2)	26.95 (2)	**17.89** (1)	42.90 (2)	**20.05** (1)	1.71
	FCRG-DT	**8.04** (1)	**4.37** (1)	**3.41** (1)	**27.02** (1)	17.77 (2)	**44.38** (1)	19.44 (2)	1.29
$\tau = 10$	SeqAttB	6.40 (4)	3.32 (4)	2.43 (4)	22.25 (4)	16.57 (4)	36.29 (4)	10.20 (4)	4.00
	HRED	6.54 (3)	3.37 (3)	2.46 (3)	22.98 (3)	17.11 (3)	37.51 (3)	15.54 (3)	3.00
	FCRG-BL	7.58 (2)	3.78 (2)	2.66 (2)	**25.09** (1)	**17.83** (1)	**39.81** (1)	**17.61** (1)	1.43
	FCRG-DT	**7.96** (1)	**3.91** (1)	**2.75** (1)	24.64 (2)	17.66 (2)	39.37 (2)	16.08 (2)	1.57

Matching and Vector Extrema. In |tokens| $\geq$ 10, our models achieved better performance than the baselines in all the embedding metrics. In particular, FCRG-BL model performed best, and then FCRG-DT model was the runner up. To sum up, FCRG-DT and FCRG-BL outperformed the baselines in Embedding metrics.

Qualitative Evaluation Next, we conducted another experiment to compare our FCRG-DT with baselines qualitatively. In the experiment, we chose FCRG-DT instead of FCRG-BL since it does not require any additional parameters and had comparable performance with FCRG-BL. We also used $\tau = 10$ to generate responses with at least 10 tokens in all models since lengthy responses are more interesting and informative despite a harder problem.

Human Evaluation Similar to [25], we randomly selected 50 original tweets from the test set. Given each of the original tweets, each of FCRG-DT, SeqAttB and HRED generated 15 responses. Then, one response with the highest probability per model was selected. We chose a pairwise comparison instead of listwise comparison to make easy for human evaluators to decide which one is better. Therefore, we created 100 triplets (original tweet, response$_1$, response$_2$) where one response was generated from our FCRG-DT and the other one was from a baseline. We employed three crowd-evaluators to evaluate each triplet where each response's model name was hidden to the evaluators. Given each triplet, the evaluators independently chose one of the following options: (i) win (response$_1$ is better), (ii) loss (response$_2$ is better), and (iii) tie (equally good or bad). Before labeling, they were trained with a few examples to comprehend the following criteria: (1) the response should fact-check information in the original tweet, (2) it should be human-readable and be free of any fluency or grammatical errors, (3) the response may depend on a specific case or may be general but do not contradict the first two criteria. The majority voting approach was employed to judge which response is better. If annotators rated a triplet with three different answers, we viewed the triplet as a tie. Table 7 shows human evaluation results. The Kappa values show moderate agreement among the evaluators. We conclude that FCRG-DT outperforms SeqAttB and HRED qualitatively.

Case Studies Table 8 presents examples of original tweets, ground-truth D-tweets, and generated responses of the three models. Our FCRG-DT generated more relevant responses with clear fact-checking intention. For example, in the first example, FCRG-DT captured the *uranium* in the original tweet and generated a relevant response. We observed that SeqAttB usually generated non-relevant content. Responses generated by FCRG-DT were more formal than ones generated by the baselines.

Table 7 The results of human evaluation

Opponent	Win	Loss	Tie	Fleiss Kappa
our FCRG-DT vs. SeqAttB	40%	28%	32%	0.725
our FCRG-DT vs. HRED	40%	36%	24%	0.592

Table 8 Examples of original tweets, ground-truth FC tweets, generated responses of our FCRG-DT and two baselines

Pairs of the original tweet (**OT**) and ground-truth **D-tweet**	Generated responses of our FCRG-DT and two baselines
OT: @user <person> sold <number> % of our uranium to russia . that's your candidate . **D-tweet:** except that she did not . url @user	**SeqAttB**: @user no , <person> did not start the birther movement . url **HRED**: @user here you go , read the whole thing . url **FCRG-DT**: @user allegations about hrc are false . she did not have to give uranium to russia : url
OT: @user i just feel bad for poor franklin (?) , all alone on his side of the table . . . **D-tweet:** @user interesting . url	**SeqAttB**: @user there is no trucker strike in puerto rico url **HRED**: @user this is a fake . sorry folks . url **FCRG-DT**: @user snopes says this is not true . url url
OT: @user that is why i asked the question so i could have an understanding **D-tweet:** @user do use a troll as your source of info url	**SeqAttB**:@user <person> kissed by former klan member : url **HRED**: @user here you go , read the whole thing . url **FCRG-DT**: @user this is a fake quote . sorry folks . url
OT: show me where <person> actually said that , other than this photoshopped picture from the internet . **D-tweet:** @user fake quote . url	**SeqAttB**: @user no, <person> did not start the birther movement. url **HRED**: @user yeah , no , they are not . url **FCRG-DT**: @user this is a fake news site , right ? url
OT: @user also , you are giving me a different set of statistics than the ones i asked for . **D-tweet:** @user here's what he misquoted to start . url	**SeqAttB**: @user here you go , read the facts : url **HRED**: @user here ya go : url read this . url **FCRG-DT**: @user this is a fake quote from <person> . url

3.4 Discussions

Although our proposed models successfully generated responses with fact-checking intention, and performed better than the baselines, there are a few limitations in our work. Firstly, we assumed guardians freely choose articles that they prefer, and then insert corresponding fact-checking URLs into our generated responses. It means we achieved partial automation in a whole fact-checking process. In our future work, we are interested in even automating the process of selecting an fact-checking article based on content of original tweets in order to fully support guardians and automate the whole process. Perhaps, combining both our recommender system and our text generation framework may help us automate the fact-checking process. Secondly, our framework is based on word-based RNNs, leading to a common issue: rare words are less likely to be generated. A feasible solution is using character-level RNNs [9] so that we do not need to replace rare words with `<unk>` token. In the future work, we will investigate if character-based RNN models work well on our dataset. Thirdly, we only used pairs of an original tweet and a D-tweet without utilizing other data sources such as previous messages in online dialogues. We also tried to use the content of fact-checking articles, but did not improve performance of our models. We plan to explore other ways to utilize the data sources in the future. Finally, there are many original tweets containing URLs pointing to fake news sources (e.g. breitbart.com) but we did not consider them when generating responses. We leave this for future exploration.

4 Conclusions

In this chapter, we presented novel preventive methods to combat fake news by leveraging online users called guardians. By identifying these guardians and analyzing their behavior in posting fact-checking tweets, we built a novel fact-checking URL recommendation model to personalize fact-checking articles and a response generation framework to help guardians fact-check information faster. In the discussion sections, we described possible extensions of our models to achieve better performance. We believe that our work opens new research directions in fake news intervention.

4.1 Contributions

Portions of this chapter are based on work that appeared in the 2018 and 2019 International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) [28, 29, 32].

References

1. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In: Proceedings of the 3rd International Web Science Conference, p. 2. ACM (2011)
2. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Semantic enrichment of twitter posts for user profile construction on the social web. In: Extended Semantic Web Conference, pp. 375–389. Springer (2011)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
4. Banerjee, S., Lavie, A.: Meteor: an automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72 (2005)
5. Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.: Short and tweet: experiments on recommending content from information streams. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1185–1194. ACM (2010)

6. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
7. Ecker, U.K., Lewandowsky, S., Tang, D.T.: Explicit warnings reduce but do not eliminate the continued influence of misinformation. Mem. Cognit. **38**(8), 1087–1100 (2010)
8. Friggeri, A., Adamic, L.A., Eckles, D., Cheng, J.: Rumor cascades. In: ICWSM (2014)
9. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. Association for the Advancement of Artificial Intelligence (AAAI) (2016)
10. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
11. Lab, R.: Fact-checking triples over four years. https://reporterslab.org/fact-checking-triples-over-four-years/ (2018)
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1188–1196 (2014)
13. Lee, K., Mahmud, J., Chen, J., Zhou, M., Nichols, J.: Who will retweet this? Automatically identifying and engaging strangers on twitter to spread information. In: Proceedings of the 19th international conference on Intelligent User Interfaces, pp. 247–256. ACM (2014)
14. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, pp. 2177–2185 (2014)
15. Liang, D., Altosaar, J., Charlin, L., Blei, D.M.: Factorization meets the item embedding: regularizing matrix factorization with item co-occurrence. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 59–66. ACM (2016)
16. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)
17. Liu, C.W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., Pineau, J.: How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2122–2132 (2016)
18. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
19. Maddock, J., Starbird, K., Al-Hassani, H.J., Sandoval, D.E., Orand, M., Mason, R.M.: Characterizing online rumoring behavior using multi-dimensional signatures. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 228–241. ACM (2015)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
21. Nyhan, B., Reifler, J.: When corrections fail: the persistence of political misperceptions. Polit. Behav. **32**(2), 303–330 (2010)
22. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
23. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
24. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. arXiv preprint arXiv:1507.04808 (2015)
25. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. arXiv preprint arXiv:1503.02364 (2015)
26. Shao, C., Ciampaglia, G.L., Flammini, A., Menczer, F.: Hoaxy: a platform for tracking online misinformation. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 745–750. International World Wide Web Conferences Steering Committee (2016)

27. Starbird, K., Palen, L.: Pass it on? Retweeting in mass emergency. In: Proceedings of the 7th International ISCRAM Conference (2010)
28. Vo, N., Lee, K.: The rise of guardians: fact-checking url recommendation to combat fake news. In: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (2018)
29. Vo, N., Lee, K.: Learning from fact-checkers: analysis and generation of fact-checking language. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–344. ACM (2019)
30. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456. ACM (2011)
31. Yang, C., Harkreader, R., Zhang, J., Shin, S., Gu, G.: Analyzing Spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In: WWW (2012)
32. You, D., Vo, N., Lee, K., Liu, Q.: Attributed multi-relational attention network for fact-checking url recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1471–1480. ACM (2019)
33. Zhao, Z., Resnick, P., Mei, Q.: Enquiring minds: Early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1395–1405. ACM (2015)