Learning from Fact-checkers: Analysis and Generation of Fact-checking Language

Nguyen Vo
Worcester Polytechnic Institute
Worcester, Massachusetts, USA, 01609
nkvo@wpi.edu

Kyumin Lee
Worcester Polytechnic Institute
Worcester, Massachusetts, USA, 01609
kmlee@wpi.edu

ABSTRACT

In fighting against fake news, many fact-checking systems comprised of 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 another complementary approach to mitigate the spread of misinformation. In this paper, we introduce a novel application of text generation for combating fake news. In particular, we (1) leverage online users named *fact-checkers*, who cite fact-checking sites as credible evidences to fact-check information in public discourse; (2) analyze linguistic characteristics of fact-checking tweets; and (3) propose and build a deep learning framework to generate responses with fact-checking intention to increase the fact-checkers' engagement in fact-checking activities. Our analysis reveals that the fact-checkers tend to refute misinformation and use formal language (e.g. few swear words and Internet slangs). Our framework successfully generates relevant responses, and outperforms competing models by achieving up to 30% improvements. Our qualitative study also confirms that the superiority of our generated responses compared with responses generated from the existing models.

ACM Reference Format:
Nguyen Vo and Kyumin Lee. 2019. 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 (SIGIR '19), July 21–25, 2019, Paris, FranceSIGIR* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3331184.3331248

1 INTRODUCTION

Our media landscape has been flooded by a large volume of falsified information, overstated statements, false claims, fauxtography and fake videos[1] perhaps due to the popularity, impact and rapid information dissemination of online social networks. The unprecedented amount of disinformation 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.

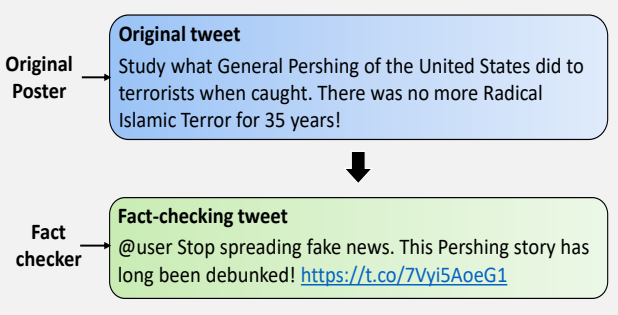


Figure 1: A real-life fact-checking activity where the fact-checker refutes misinformation in the original tweet.

To fight against fake news, many fact-checking systems ranging from human-based systems (e.g. Snopes.com), classical machine learning frameworks [20, 34, 38] to deep learning models [29, 39, 56, 57] were developed to determine credibility of online news and information. However, falsified news is still disseminated like wild fire [31, 59] despite dramatic rise of fact-checking sites worldwide [21]. Furthermore, recent work showed that individuals tend to selectively consume news that have ideologies similar to what they believe while disregarding contradicting arguments [8, 35]. These reasons and problems indicate that using only fact-checking systems to debunk fake news is insufficient, and complementary approaches are necessary to combat fake news.

Therefore, in this paper, we focus on online users named *fact-checkers*, who directly engage with other users in public dialogues and convey verified information to them. Figure 1 shows a real-life conversation between a user, named *original poster*, and a fact-checker. In Figure 1, the original poster posts a false claim related to General Pershing. The *fact-checker* 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 *fact-checking tweet* (FC-tweet). Recent work [52] showed that fact-checkers often quickly fact-checked original tweets within a day after being posted and their FC-tweets could reach hundreds of millions of followers. Additionally, [9] showed that the likelihood to delete shares of fake news increased by four times when there existed a fact-checking URL in users' comments. In our analysis, we also observe that after receiving FC-tweets, 7% original tweets were not accessible because of account suspension, tweet deletion, and a private mode.

Due to the fact-checkers' activeness and high impact on dissemination of fact-checked content, in this paper, our goal is to further

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

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331248

support them in fact-checking activities toward complementing existing fact-checking systems and combating fake news. In particular, we aim to build a text generation framework to generate responses[2] 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. We assume that fact-checkers choose the fact-checking URLs by themselves based on their interests (e.g., https://t.co/7Vyi5AoeG1 in Figure 1). Therefore, we focus on generating responses without automatically choosing specific fact-checking URLs, which is beyond the scope of this paper.

To achieve the goal, we have to solve the following research problems: (**P1**) how can we obtain a dataset consisting of original tweets and associated fact-checking replies (i.e., replies which exhibit fact-checking intention)?; (**P2**) how can we analyze how fact-checkers communicate fact-checking content to original posters?; and (**P3**) how can we automatically generate fact-checking responses when given content of original tweets?

To tackle the first problem **(P1)**, we may use already available datasets [15, 52, 54]. However, the dataset in [15] contains relatively small number of original tweets (~5,000) and many FC-tweets (~170K). Since FC-tweet generation process depends on contents of original tweets, it may reduce diversity of generated responses. The dataset in [54] is large but fully anonymized, and the dataset in [52] does not contain original tweets. Therefore, we collected our own dataset consisting of 64,110 original tweets and 73,203 FC-tweets (i.e., each original tweet receives 1.14 FC-tweet) by using Hoaxy system [46] and FC-tweets in [52].

To understand how fact-checkers convey credible information to original posters and other users in online discussions (**P2**), we conducted data-driven analysis of FC-tweets and found that fact-checkers tend to refute misinformation and employ more impersonal pronouns. Their FC-tweets were generally more formal and did not contain much swear words and Internet slangs. These analytical results are important since we can reduce the likelihood to generate racist tweets [14], hate speeches [7] and trolls [4].

To address the third problem (**P3**), we propose a deep learning framework to automatically generate fact-checking responses for fact-checkers. In particular, we build the framework based on Seq2Seq [49] with attention mechanisms [28].

Our contributions are as follows:

- To the best of our knowledge, we are the first to propose a novel application of text generation for supporting fact-checkers and increasing their engagement in fact-checking activities.
- We conduct a data-driven analysis of linguistic dimensions, lexical usage and semantic frames of fact-checking tweets.
- We propose and build a deep learning framework to generate responses with fact-checking intention. Experimental results show that our models outperformed competing baselines quantitatively and qualitatively.
- We release our collected dataset and source code in public to stimulate further research in fake news intervention[3].

[2]We use the term "fact-checking tweets (FC-tweets)", "fact-checking responses", and "fact-checking replies" interchangeably.
[3]https://github.com/nguyenvo09/LearningFromFactCheckers

2 RELATED WORK

In this section, we briefly cover related works about (1) misinformation and fact-checking, and (2) applications of text generation.

2.1 Misinformation and Fact-checking

Fake news is recently emerging as major threats of credibility of information in cyberspace. Since human-based fact-checking sites could not fact-check every falsified news, many automated fact-checking systems were developed to detect fake news in its early stage by using different feature sets [10, 15, 40, 54, 59], knowledge graph [47] and crowd signals [16, 17, 34], and using deep learning models [30, 39, 57]. In addition, other researchers studied how to fact-check political statements [56], mutated claims from Wikipedia [50] and answers in Q&A sites [32].

Other researchers studied intention of spreading fake news (e.g. misleading readers, inciting clicks for revenue and manipulating public opinions) and different types of misinformation (e.g. hoaxies, clickbait, satire and disinformation) [42, 53]. Linguistic patterns of political fact-checking webpages and fake news articles [13, 42] were also analyzed. Since our work utilizes FC-tweets, analyzing users' replies [9, 15, 40, 54] are closely related to ours. However, the prior works had limited attention on analyzing how fact-checkers convey fact-checked content to original posters in public discourse.

Additionally, researchers investigated topical interests and temporal behavior of fact-checkers [52], relationship between fact-checkers and original posters [11], how fake news disseminated when fact-checked evidences appeared [9], and whether users were aware of fact-checked information when it was available [15]. Our work is different from these prior works since we focus on linguistic dimensions of FC-tweets, and propose and build a response generation framework to support fact-checkers.

2.2 Applications of Text Generation

Text generation has been used for language modeling [33], question and answering [12], machine translation [1, 28, 49], dialogue generation [43–45, 51, 55], and so on. Recently, it is employed to build chat bots for patients under depression[4], customer assistants in commercial sites, teen chat bots [14], and supporting tools for teachers [3]. Text generation has been also used to detect fake review [58], clickbait headlines [48], and fake news [41]. Our study is the first work that generates responses based on FC-tweets as a supporting tool for fact-checkers. Our work is closely related with dialog generation in which there are three main technical directions: (1) deterministic models [43, 45, 51, 55], (2) Variational Auto-Encoders (VAEs) [44], and (3) Generative Adversarial Networks (GANs) [25]. Although recently VAEs and GANs showed promising results, deterministic models are still dominant in literature since they are easier to train than VAEs and GANs, and achieve competitive results [22]. Thus, we propose a response generation framework based on Seq2Seq and attention mechanism [28].

3 DATASET

In this section, we describe our data collection and preprocessing process. We utilized the dataset in [52] and the Hoaxy system [46]

[4]https://read.bi/2QZ0ZPn

to collect FC-tweets, which contained fact-checking URLs from two popular fact-checking sites: *snopes.com*, and *politifact.com*. Totally, we collected 247,436 distinct fact-checking tweets posted between May 16, 2016 and May 26, 2018.

Similar to [11, 52], we removed non-English FC-tweets, and FC-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 fact-checking pages, 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 fact-checking tweet. As a result, 190,158 FC-tweets and 164,477 distinct original tweets were remained.

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

4 LINGUISTIC CHARACTERISTICS OF FACT-CHECKING TWEETS

Since our goal is to automatically generate responses with fact-checking intention, it is necessary to analyze what kind of linguistic characteristics FC-tweets have, and verify whether FC-tweets in our dataset have the fact-checking intention.

To highlight linguistic characteristics of FC-tweets, we compare FC-tweets with *Normal Replies*, which are direct responses to the same 64,110 original tweets without including fact-checking URLs, and *Random Replies*, which do not share any common original tweets with FC-tweets. Initially, we collected 262,148 English *Normal Replies* and 97M English *Random Replies* posted in the same period of the FC-tweets. Then, we sampled 73,203 Normal Replies and 73,203 Random Replies from the initial collection to balance the data with our FC-tweets. All of the FC-tweets, Normal Replies and Random Replies were firstly preprocessed by replacing URLs with *url* and mentions with *@user*, and by removing special characters. They were tokenized by the NLTK tokenizer. Then, we answer the following research questions. Note that we sampled 73,203 Random Replies and 73,203 Normal Replies four times more, and our analysis was consistent with the following results.

Q1: What are underlying themes in FC-tweets?

To answer this question, we applied the standard LDA algorithm to each of the three types of replies, so we built three independent LDA models. Table 1 shows 5 topics extracted from each of the three LDA topic models with associated keywords. Firstly, FC-tweets exhibit clear fact-checking intention with keywords such as debunked, snopes, read, stop, check, and lie. Secondly, keywords of Normal Replies show awareness of misinformation. However, fact-checking intention is not clear compared with FC-tweets. The keywords of Random Replies are commonly used in daily conversations. Based on the analysis, we conclude that the main themes of FC-tweets are about fact-checking information in the original tweets.

Table 1: 5 LDA topics and associated keywords of FC-tweets, Normal Replies and Random Replies.

Types	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
FC-tweets	read stop try	fake news that's	facts check debunked	false snopes lie	true know story
Normal Replies	one liar know	u like f***	fake news trump	president trump get	like really stop
Random Replies	love u know	good thanks yes	like one people	like would right	thank oh im

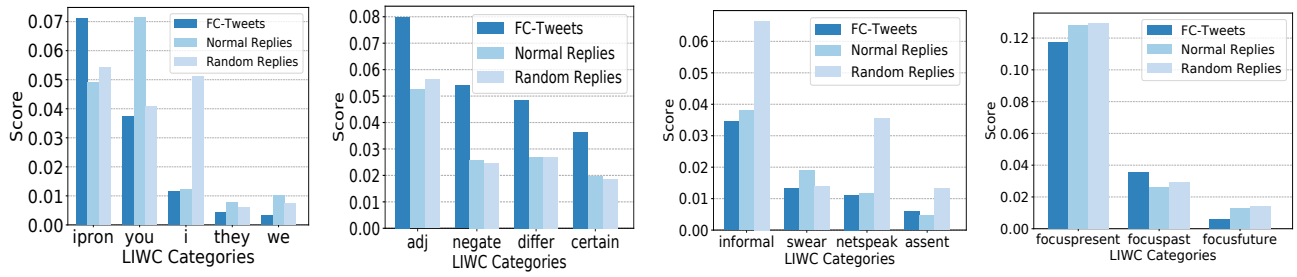
Q2: What are the psycholinguistic characteristics of FC-tweets?

We employed LIWC 2015 [37], a standard approach for mapping text to 73 psychologically-meaningful categories, for understanding psychological characteristics of FC-tweets. Given each of FC-tweets, we counted how many words of the FC-tweet belonged to each LIWC category. Then, we computed a normalized score for the category by diving the count by the number of words in the FC-tweet. Finally, we report the average scores μ and variances σ^2 for each LIWC category based on |FC-tweets|. The same process was applied to Normal Replies and Random Replies. We examined all LIWC categories and report only the most significant results.

(A) FC-tweets have the highest usage of impersonal pronouns and the least utilization of personal pronouns. In Figure 2(a), we can see that FC-tweets exhibit the highest usage of impersonal pronouns (e.g. it, this, that) ($\mu = 0.071, \sigma^2 = 0.01$) in comparison with Normal Replies ($\mu = 0.049, \sigma^2 = 0.009$) and Random Replies ($\mu = 0.054, \sigma^2 = 0.005$). This observation is statistically significant in Mann Whitney one sided U-test ($p < 0.001$). Examples of FC-tweets containing impersonal pronouns (called *iprons* in LIWC) are (i) *@user **This** has been debunked repeatedly - url please stop spreading the lie, thanks!*, and (ii) *@user **it** is a wonderful quote but Lewis never said it : url and url.*

Differently, Normal Replies show the highest mean score in 2nd person pronouns (named *you* category) ($\mu = 0.071, \sigma^2 = 0.007, p < 0.001$) in comparison to FC-tweets ($\mu = 0.037$) and Random Replies ($\mu = 0.047$). Note that *you* in this context may refer to the original posters. In 1st person pronouns (named *I* category), Random Replies have highest score because they contain daily personal conversations between online users. Finally, FC-tweets have the smallest usage of *we* ($\mu = 0.003, \sigma^2 = 0.000$) and *they* ($\mu = 0.004, \sigma^2 = 0.000$) among three groups of replies ($p < 0.001$).

(B) FC-tweets have a tendency to refute content of original tweets. Figure 2(b) shows the mean scores of *adj, negate, differ* and *certain* categories. Specifically, FC-tweets exhibit the highest mean score in *adjectives* category ($\mu = 0.080, \sigma^2 = 0.024, p < 0.001$) in comparison to Normal Replies ($\mu = 0.052, \sigma^2 = 0.008$) and Random Replies ($\mu = 0.056, \sigma^2 = 0.015$). Prevalent adjectives in FC-tweets are fake, wrong, dump, false, and untrue. FC-tweets also tend to refute information of original tweets. Their mean score in *negate* category is 0.054, which is about two times higher than the mean score of Normal Replies ($p < 0.001$). FC-tweets have also



(a) Pronouns (b) Negation (c) Informality (d) Temporal Dimensions

Figure 2: LIWC category scores of FC-tweets, Normal Replies and Random Replies. In the figure, we follow LIWC category abbreviations as labels [37]. For example, *ipron* and *adj* mean impersonal pronouns and adjectives, respectively.

the highest usage of words in *differ* category (e.g. actually, but, except) among the three groups ($p < 0.001$). In *certain* category (e.g. never, ever, nothing, always), FC-tweets' mean score ($\mu = 0.036$) also doubles the average score of Normal Replies significantly ($p < 0.001$). Examples of FC-tweets are: (i) *@user* ***wrong***. ***never*** *happened. url*, (ii) *@user except he* ***didn't***. *that tweet has been proven* ***fake****: url*, and (iii) *@user I* ***sure*** *hope you're joking. url.*

(C) FC-tweets are usually more formal and have low usage of swear words. In Figure 2(c), FC-tweets have lower mean score in *informal* category ($\mu = 0.034, \sigma^2 = 0.010$) than Normal Replies ($\mu = 0.038, p < 0.001$) and Random Replies ($\mu = 0.066$). FC-tweets also use the least swear words ($\mu = 0.013, \sigma^2 = 0.005, p < 0.005$) among the three groups. In terms of *netspeak* category (i.e. Internet slangs), FC-tweets ($\mu \approx 0.011$) generally have smaller average score than Random Replies ($\mu = 0.035$). Furthermore, FC-tweets do not contain much words in *assent* category (e.g. OK, yup, okey) ($\mu = 0.006, \sigma^2 = 0.002$) compared with Random Replies ($\mu = 0.013, \sigma^2 = 0.005, p < 0.001$). Regarding formality of FC-tweets, we conjecture that fact-checkers try to persuade original posters to stop spreading fake news, leading to more formal language, less usage of swear words. An example of FC-tweets is *@user url I'm sure you'll still say it's true- but it simply isn't. Google for facts and debunks please.*

(D) FC-tweets emphasize on what happened in the past whereas Normal Replies and Random Replies focus on present and future. In Figure 2(d), FC-tweets usually employ verbs in past tense to mention past stories to support their factual corrections. Thus, the average score of *focuspast* category of FC-tweets is the highest ($\mu = 0.036, \sigma^2 = 0.005, p < 0.001$) among the three groups of replies, whereas Normal Replies and Random Replies emphasize on present and future. Particularly, FC-tweets have the least score in *focuspresent* ($\mu = 0.117, \sigma^2 = 0.015, p < 0.001$) and *focusfuture* ($\mu = 0.006, \sigma^2 = 0.000, p < 0.001$) categories. An example of FC-tweets is *@user yeah, she merely* ***said*** *something that* ***was*** *only slightly less absurd. url.*

Q3: How are semantic frames represented in FC-tweets?

So far, we examined every word independently without considering its dependencies with other words (e.g. surrounding words), which is helpful in understanding its meaning. Thus, we now employ SEMAFOR model [6], trained on FrameNet data[5], to extract rich structures called *semantic frames* based on syntactic trees of

[5]https://framenet.icsi.berkeley.edu/fndrupal/frameIndex

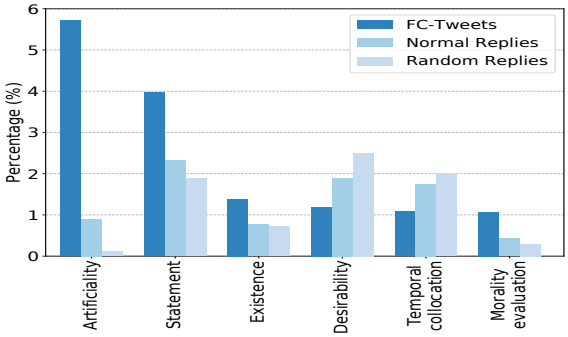
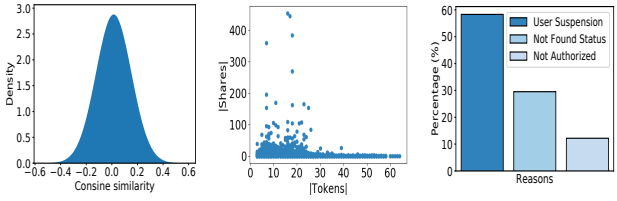


Figure 3: Distribution of semantic frame types in FC-tweets.

sentences. For example, a frame *Statement* consists of *a noun* and *a verb* where *the noun* indicates a speaker and *the verb* implies the act of conveying a message. We measured the distribution of semantic frames of FC-tweets by firstly counting the number of occurrences of every frame type across all FC-tweets, and normalized it by the total number of detected frames in all FC-tweets. The same process was applied to Normal Replies and Random Replies. Figure 3 shows the percentage of different types of frames detected by SEMAFOR. We have the following observations:

(A) FC-tweets display high usage of *Artificiality*, *Statement* and *Existence*. In Figure 3, FC-tweets have the highest utilization of *Artificiality* (e.g. wrong, lie, fake, false, genuine, phoney) among three groups of replies ($p < 0.001$ according to one-sided z-test). This frame accounts for 5.71% detected frames in FC-tweets compared with Normal Replies (0.90%) and Random Replies (0.11%). FC-tweets also have the highest proportion of the frame *Statement* (3.96%, $p < 0.001$) among three groups of replies. Words that evoke frame *Statement* in FC-tweets are said, says, claims, report, told, talk, and mention. Examples of FC-tweets are (i) *@user You're the one who has no clue. She never* ***said*** *this: url*, and (ii) *@user Snopes* ***reports*** *this rumor as* ***false****. url.* To refer to verified information, FC-tweets employed frame *Existence* (1.38%, $p < 0.001$) compared with Normal Replies (0.71%) and Random Replies (0.70%). The most popular phrases evoking frame *Existence* were real, there is, there are, exist, there were, and there have been. Examples of FC-tweets are: (i) *@user* ***There is*** *no trucker strike in Puerto Rico url*, (ii) *@user That town doesnt* ***exist*** *url.*



(a) FC-tweets vs. articles (b) |Shares| vs. |tokens| (c) Irretrievable tweets

Figure 4: (a) Similarity between FC-tweets and fact-checking articles, (b) |Shares| vs. |Tokens| of FC-tweets and (c) Distribution of irretrievable original tweets.

(B) FC-tweets exhibit the highest *Morality_evaluation* and have less usage of *Desirability*. As shown in Figure 3, FC-tweets contain the highest proportion of frame *Morality_evaluation* (1.06%, $p < 0.001$) among three groups. The most popular words in frame *Morality_evaluation* are wrong, evil, dishonest, despicable, unethical, and immoral. Another supporting evidence of this observation is the lower usage of frame *Desirability* (e.g. good, better, bad, great, okay, cool) in FC-tweets (1.19%, $p < 0.001$) than Normal Replies (1.89%) and Random Replies (2.5%). An example of FC-tweets is: *@user you're such an **evil, despicable** creature. url*

(C) FC-tweets do not use much *Temporal_collocation*. FC-tweets show lower usage of *Temporal_collocation* (1.08%, $p < 0.001$) than Normal Replies (1.74%) and Random Replies (2.00%). The most common words that evoke this frame in FC-tweets are when, now, then, today, current, recently, future. It seems these words are mainly about the present and the future. This result supports the same observation that FC-tweets tend to focus on the past.

Q4: Do fact-checkers include details of fact-checking articles? We firstly derived latent representations of FC-tweets and articles by training two Doc2Vec models [23] - one for FC-tweets and the other one for fact-checking articles. The embedding size is 50. Then, we measured cosine similarity between a FC-tweet and the fact-checking article embedded in the FC-tweet as shown in Figure 4(a). Interestingly, most FC-tweets do not have high similarity with FC-articles, suggesting that fact-checkers rarely include details from fact-checking articles in FC-tweets. However, there were several enthusiastic fact-checkers who extracted information from fact-checking articles to make FC-tweets more persuasive, as shown in two tails of the curve in Figure 4(a).

Q5: Is there any connection between |tokens| of FC-tweets and |shares|? Since sharing FC-tweets by retweets and quotes is important for increasing the visibility of credible information on online social networks, we examined correlation between |tokens| of FC-tweets and their |shares|. We only focus on |tokens| because it could help us to decide length of a generated response. Figure 4(b) shows a scatter plot of FC-tweets' |tokens| and |shares| (i.e., quotes and retweets). Generally, most FC-tweets had |shares|=0. However, FC-tweets with |tokens| $\in [10; 20]$ usually received more attention. To verify this, we created two lists – one containing |shares| of FC-tweets with |tokens| $\in [0; 9]$ and another one for |shares| of FC-tweets with |tokens| $\in [10; 20]$ –, and then conducted Mann Whitney one-sided U-test. We found that the latter one had significantly larger numbers than the former one ($p = 2.91 \times 10^{-11}$). We conclude that very short FC-tweets may be not informative enough to draw readers' attention, and lengthy FC-tweets may be too time-consuming to read, leading to small number of shares. Therefore, a reasonable length of FC-tweets is more preferable when we generate a response.

Q6: Is there any signal suggesting positive effect of FC-tweets? We examined what happened to original tweets after receiving fact-checking tweets. In Oct 2018 (i.e., five months after collecting our dataset), we re-collected original tweets via Twitter APIs to see if all of the original tweets were retrievable. Interestingly, 4,516 (7%) original tweets were not retrievable. There are three reasons: (i) *User Suspension*, (ii) *Not Found Status* (i.e. deleted status), and (iii) *Not Authorized* (i.e. original tweets are in the private mode).

In Figure 4(c), *User Suspension* accounted for 58.30% of the irretrievable original tweets. Although there may be many factors that potentially explain suspension (e.g. original posters may have other abusing behaviors that triggered Twitter security system), one obvious observation is that fact-checkers tended to target bad users (e.g. content polluters [24]), who usually have abusing behavior on social platform. It means that fact-checkers are enthusiastic about checking credibility of information on social networks. Regarding two reasons *Not Authorized* and *Not Found Status*, perhaps original posters were either aware of the wrong information they posted or were under pressure due to criticisms they received from other users, leading to deletion or hiding their original tweets.

In summary, our analysis reveals fact-checkers refuted content of original tweets, and their FC-tweets were more formal than Normal Replies and Random Replies. To provide supporting evidences, FC-tweets utilized semantic frames *Existence* and *Statement*. These results confirm that FC-tweets exhibit clear fact-checking intention.

5 RESPONSE GENERATION FRAMEWORK

In the previous section, we analyzed common topics, lexical usages, and distinguishing linguistic dimensions of FC-tweets compared with Normal Replies and Random Replies. Our analysis revealed that FC-tweets indeed exhibited clear fact-checking intention, which is the property that we desired. Now, we turn our attention to proposing and building our framework, named *Fact-checking Response Generator* (**FCRG**), in order to generate responses with fact-checking intention. The generated responses are used to support fact-checkers and increase their engagement.

Formally, given a pair of an original tweet and a FC-tweet, the original tweet x is a sequence of words $x = \{x_i | i \in [1; N]\}$ and the FC-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 FC-tweet, respectively. We inserted a special token <s> as a starting token into every FC-tweet. Drawing inspiration from [28], we propose and build a framework as shown in Figure 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 FC-tweet. Their details are as follows:

5.1 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

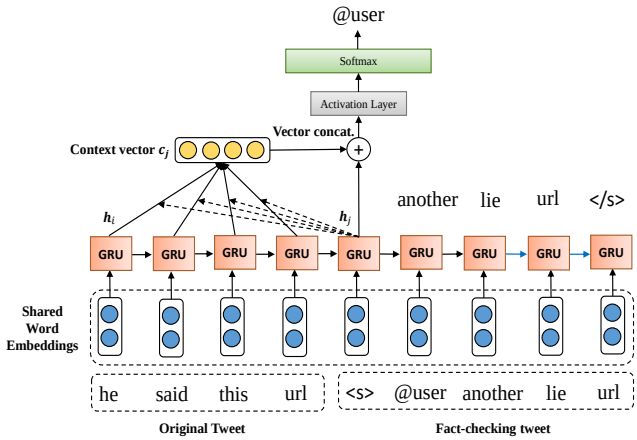


Figure 5: Our proposed framework to generate responses with fact-checking intention.

an embedding matrix and V is the vocabulary size. We use the same word embedding matrix $\mathbf{W}_e$ for the FC-tweet. In particular, for every word y_i (represented as one-hot vector $y_i \in \mathbb{R}^V$) in the FC-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. 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 [28]. This is helpful in reducing overfitting.

5.2 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) [5]. Formally, we compute hidden state $\mathbf{h}_i \in \mathbb{R}^H$ at time-step i^{th} in the encoder as follows:

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

where the GRU is defined by the following equations:

$$\begin{aligned} \mathbf{z}_i &= \sigma(\mathbf{x}_i \mathbf{W}_z + \mathbf{h}_{i-1} \mathbf{U}_z) \\ \mathbf{r}_i &= \sigma(\mathbf{x}_i \mathbf{W}_r + \mathbf{h}_{i-1} \mathbf{U}_r) \\ \tilde{\mathbf{h}}_i &= tanh(\mathbf{x}_i \mathbf{W}_o + (\mathbf{r}_i \odot \mathbf{h}_{i-1}) \mathbf{U}_o) \\ \mathbf{h}_i &= (1 - \mathbf{z}_i) \odot \tilde{\mathbf{h}}_i + \mathbf{z}_i \odot \mathbf{h}_{i-1} \end{aligned} \tag{2}$$

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 $\mathbf{X} = [\mathbf{h}_1 \oplus \mathbf{h}_2 \oplus ... \oplus \mathbf{h}_N] \in \mathbb{R}^{H \times N}$, where $\oplus$ denotes concatenation of hidden states. We use the last hidden state $\mathbf{h}_N$ as features of the original tweet $\mathbf{x} = \mathbf{h}_N$.

5.3 Decoder

The decoder takes $\mathbf{x}$ as the input to start the generation of a FC-tweet. We use another GRU to represent the decoder to generate a sequence of tokens $y = \{y_1, y_2, .., y_M\}$. At each time-step j^{th}, 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 j^{th} word in the FC-tweet and the i^{th} output from the encoder. In this study, $\mathbf{a}_j$ is computed by one of the following ways:

$$\mathbf{a}_j = \begin{cases} softmax(\mathbf{X}^T \mathbf{h}_j) & \text{Dot Attention} \\ softmax(\mathbf{X}^T \mathbf{W}_a \mathbf{h}_j) & \text{Bilinear Attention} \end{cases} \tag{3}$$

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 [1] and concat attention [28] but the above attention mechanisms in Eq. 3 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 j^{th} time-step is predicted by a softmax classifier:

$$\hat{\mathbf{y}}_j = softmax\big(\mathbf{W}_s \tanh\left(\mathbf{W}_c[\mathbf{c}_j^T \oplus \mathbf{h}_j^T]^T\right)\big) \tag{4}$$

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:

$$p(y_j = v_k | y_{j-1}, y_{j-2}, ..., y_1, \mathbf{x}) = \hat{\mathbf{y}}_{jk} \tag{5}$$

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

$$p(y|x) = \prod_{j=1}^{M} p(y_j | y_{j-1}, y_{j-2}, ..., y_1, \mathbf{x}) \tag{6}$$

Since the entire architecture is differentiable, we jointly train the whole network with Teacher Forcing via Adam optimizer [19] by minimizing the negative conditional log-likelihood for m pairs of the original tweet $x^{(i)}$ and the FC-tweet $y^{(i)}$ as follows:

$$\min_{\theta_e, \theta_d} \mathcal{L} = -\sum_{i=1}^{m} \log p(y^{(i)} | x^{(i)}; \theta_e, \theta_d) \tag{7}$$

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 FC-tweet is ended when an end-of-sentence token (e.g. </s>) is emitted.

6 EVALUATION

In this section, we thoroughly evaluate our models namely **FCRG-DT** (based on dot attention in Eq. 3) and **FCRG-BL** (based on bilinear attention in Eq. 3) quantitatively and qualitatively. We seek to answer the following questions:

Table 2: Performance of our models and baselines.

Constraints	Model	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	Greedy Mat.	Vector Ext.	Avg. Rank
Without Constraints	SeqAttB	7.148 (4)	4.050 (4)	3.261 (3)	26.474 (3)	17.659 (3)	43.566 (3)	15.837 (4)	3.43
	HRED	7.301 (3)	4.073 (3)	3.248 (4)	26.222 (4)	17.545 (4)	42.734 (4)	18.929 (3)	3.57
	our FCRG-BL	**7.678** (1)	4.270 (2)	3.406 (2)	27.142 (2)	**17.871** (1)	43.714 (2)	**20.244** (1)	1.57
	our FCRG-DT	7.641 (2)	**4.303** (1)	**3.500** (1)	**27.352** (1)	17.750 (2)	**45.302** (1)	19.993 (2)	1.43
At least 5 tokens	SeqAttB	7.470 (4)	4.085 (4)	3.175 (4)	26.169 (4)	17.719 (3)	41.038 (4)	14.686 (4)	3.86
	HRED	7.631 (3)	4.155 (3)	3.227 (3)	26.241 (3)	17.617 (4)	41.930 (3)	18.850 (3)	3.14
	our FCRG-BL	7.925 (2)	4.285 (2)	3.295 (2)	26.953 (2)	**17.885** (1)	42.899 (2)	**20.052** (1)	1.71
	our FCRG-DT	**8.043** (1)	**4.373** (1)	**3.409** (1)	**27.020** (1)	17.770 (2)	**44.379** (1)	19.441 (2)	1.29
At least 10 tokens	SeqAttB	6.398 (4)	3.319 (4)	2.434 (4)	22.250 (4)	16.568 (4)	36.298 (4)	10.198 (4)	4.00
	HRED	6.540 (3)	3.373 (3)	2.462 (3)	22.980 (3)	17.106 (3)	37.513 (3)	15.537 (3)	3.00
	our FCRG-BL	7.576 (2)	3.780 (2)	2.660 (2)	**25.086** (1)	**17.832** (1)	**39.809** (1)	**17.605** (1)	1.43
	our FCRG-DT	**7.955** (1)	**3.914** (1)	**2.751** (1)	24.635 (2)	17.662 (2)	39.374 (2)	16.081 (2)	1.57

- **RQ1:** What are the performance of our models and baselines in word overlap-based metrics (i.e., measuring syntactic similarity between a ground-truth FC-tweet and a generated one)?
- **RQ2:** How do our models perform compared with baselines in embedding metrics (i.e., measuring semantic similarity between a ground-truth FC-tweet and a generated one)?
- **RQ3:** How does the number of generated tokens of responses affect performance of our models?
- **RQ4:** Is our generated responses better than ones generated from baselines in a qualitative evaluation?
- **RQ5:** What word embedding representatives in our model are close to each other in the semantic space?

6.1 Baselines and Our Models

Since our methods are deterministic models, we compare them with state-of-the-art baselines in this direction.

- **SeqAttB:** Shang et al. [45] proposed a hybrid model that combines global scheme and local scheme [1] 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 [43] 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.

6.2 Experimental Settings

Data Processing. Similar to [43] 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 FC-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 FC-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 FC-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 (as we learned in Section 4), we forced the beam search to generate responses with at least τ tokens. Since 92.4% of FC-tweets had |tokens| ≥ 5, and 60% FC-tweets had |tokens| ≥ 10, we chose $\tau \in \{0, 5, 10\}$. Moreover, as shown in Figure 4(b), FC-tweets with |tokens| $\in [10; 20]$ usually had more shares than FC-tweets with |tokens| < 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 [36], ROUGE-L [26], and METEOR [2]. These metrics evaluate the amount of overlapping words between a generated response and a ground-truth FC-tweet. The higher score indicates that the generated response are close/similar to the ground-truth FC-tweet syntactically. In other words, the generated response and the FC-tweet have a large number of overlapping words. Additionally, we also used embedding metrics (i.e. Greedy Matching and Vector Extrema) [27]. 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 FC-tweet are compared by a measure such as cosine similarity. The higher value means the response and the FC-tweet are semantically similar.

Table 3: 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

6.3 RQ1 & RQ3: 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 2 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 i^{th} 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 2. It makes sense because generating longer response similar with a ground-truth FC-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.

6.4 RQ2 & RQ3: Quantitative Results based on Embedding Metrics

We adopted two embedding metrics to measure semantic similarity between generated responses and ground-truth FC-tweets [27]. Again, we tested all the models under three settings as shown in Table 2. 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 Matching and Vector Extrema. In |tokens| ≥ 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.

6.5 RQ4: 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 [45], 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 3 shows human evaluation results. The Kappa values show moderate agreement among the evaluators. We conclude that FCRG-DT outperforms SeqAttB and HRED qualitatively.

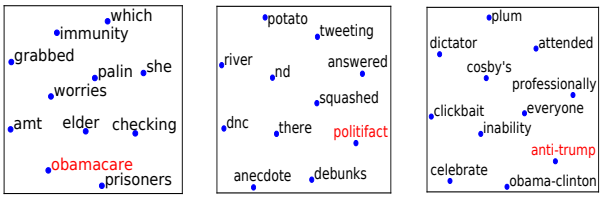
Case Studies. Table 4 presents examples of original tweets, ground-truth FC-tweets, and generated responses of the three models. Our FCRG-DT generated more relevant responses with clear fact-checking intention. For example, in the first example, FCRG-DT captured the *uranium* in the original tweet and 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.

6.6 RQ5: Similar Words in the Semantic Space

As our word embeddings were trained from scratch, we seek to investigate if our model can identify words semantically close/similar to each other. This analysis will help us gain more insights about

Table 4: 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 **FC-tweet**	Generated responses of our FCRG-DT and two baselines
OT: @user <person> sold <number> % of our uranium to russia . that's your candidate . **FC-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 ... **FC-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 **FC-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 . **FC-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 . **FC-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



(a) obamacare (b) politifact (c) anti-trump

Figure 6: 2-D visualization of word embeddings learned from our FCRG-DT model.

our dataset. After training our FCRG-DT, we extracted the embedding vectors from the shared embedding layer shown in Figure 5. We selected three keywords such as *obamacare*, *politifact* and *anti-trump*. For each keyword, we found top 10 most similar words based on cosine similarity between extracted embedding vectors and used t-SNE to project these vectors into 2D space as shown in Figure 6. Firstly, the keyword *obamacare* associated with health care policies of the Obama administration in Figure 6(a) was similar with *elder*, *AMT* (Alternative Minimum Tax), *immunity* and *checking*. Next, *politifact* in Fig. 6(b) is close to *debunks*, *there*, *anecdote*, *DNC* (Democratic National Committee) and *answered*. Finally, with *anti-trump* in Fig. 6(c), our model identified *obama-clinton*, *dictator*, *clickbait*, and *inability*. Based on this analysis, we conclude that embedding vectors learned from our model can capture similar words in the semantic space.

7 DISCUSSION

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 fact-checkers are 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 fact-checkers and automate the whole 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 [18] 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 a original tweet and a FC-tweet without utilizing other data sources such as previous messages in online dialogues. As we showed in Figure 4(a), FC-tweets often did not contain content of fact-checking articles, leading to difficulties in using this data source. We 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.

8 CONCLUSIONS

In this paper, we introduced a novel application of text generation in combating fake news. We found that there were distinguishing linguistic features of FC-tweets compared with Normal and Random Replies. Notably, fact-checkers tended to refute information in original tweets and referred to evidences to support their factual corrections. Their FC-tweets were usually more formal, and contained less swear words and Internet slangs. These findings indicate that fact-checkers sought to persuade original posters in order to stop spreading fake news by using persuasive and formal language. In addition, we observed that when FC-tweets were posted, 7% original tweets were removed, deleted or hidden from the public. After analyzing FC-tweets, we built a deep learning framework to generate responses with fact-checking intention. Our model FCRG-DT was able to generate responses with fact-checking intention and outperformed the baselines quantitatively and qualitatively. Our work has opened a new research direction to combat fake news by supporting fact-checkers in online social systems.

ACKNOWLEDGMENT

This work was supported in part by NSF grant CNS-1755536, AWS Cloud Credits for Research, and Google Cloud. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR* (2015).
[2] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL*.
[3] Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. 2018. LearningQ: a large-scale dataset for educational question generation. In *ICWSM*.
[4] J Cheng, M Bernstein, C Danescu-Niculescu-Mizil, and J Leskovec. 2017. Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions.. In *CSCW*.
[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
[6] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *NAACL*. 948–956.
[7] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009* (2017).
[8] Ullrich KH Ecker, Stephan Lewandowsky, and David TW Tang. 2010. Explicit warnings reduce but do not eliminate the continued influence of misinformation. *Memory & cognition* 38, 8 (2010), 1087–1100.
[9] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor Cascades.. In *ICWSM*.
[10] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. 2013. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *WWW*. ACM, 729–736.
[11] Aniko Hannak, Drew Margolin, Brian Keegan, and Ingmar Weber. 2014. Get Back! You Don't Know Me Like That: The Social Mediation of Fact Checking Interventions in Twitter Conversations.. In *ICWSM*.
[12] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*. 1693–1701.
[13] Benjamin D Horne and Sibel Adali. 2017. This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *NECO Workshop* (2017).
[14] Business Insider. 2016. Microsoft is deleting its AI chatbot's incredibly racist tweets. https://read.bi/2DgeRkN. (2016).
[15] Shan Jiang and Christo Wilson. 2018. Linguistic Signals under Misinformation and Fact-Checking: Evidence from User Comments on Social Media. *HCI* (2018).
[16] Jooyeon Kim, Dongkwan Kim, and Alice Oh. 2019. Homogeneity-Based Transmissive Process to Model True and False News in Social Networks. In *WSDM*.
[17] Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2018. Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation. In *WSDM*.
[18] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-Aware Neural Language Models.. In *AAAI*.
[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[20] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Aspects of rumor spreading on a microblog network. In *SocInfo*.
[21] Reporter Lab. 2018. Fact-checking triples over four years. https://reporterslab.org/fact-checking-triples-over-four-years/. (2018).
[22] Hung Le, Truyen Tran, Thin Nguyen, and Svetha Venkatesh. 2018. Variational memory encoder-decoder. In *NIPS*.
[23] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*.
[24] Kyumin Lee, James Caverlee, and Steve Webb. 2010. Uncovering social spammers: social honeypots+ machine learning. In *SIGIR*.
[25] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*.
[26] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).
[27] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023* (2016).
[28] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.
[29] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks.. In *IJCAI*. 3818–3824.
[30] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *CIKM*.
[31] Jim Maddock, Kate Starbird, Haneen J Al-Hassani, Daniel E Sandoval, Mania Orand, and Robert M Mason. 2015. Characterizing online rumoring behavior using multi-dimensional signatures. In *CSCW*.
[32] Tsvetomila Mihaylova, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass. 2018. Fact checking in community forums. In *AAAI*.
[33] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *ISCA*.
[34] An T Nguyen, Aditya Kharosekar, Matthew Lease, and Byron Wallace. 2018. An interpretable joint graphical model for fact-checking from crowds. In *AAAI*.
[35] Brendan Nyhan and Jason Reifler. 2010. When corrections fail: The persistence of political misperceptions. *Political Behavior* 32, 2 (2010), 303–330.
[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.
[37] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.
[38] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2016. Credibility assessment of textual claims on the web. In *CIKM*.
[39] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *EMNLP*.
[40] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *EMNLP*.
[41] Feng Qian, ChengYue Gong, Karishma Sharma, and Yan Liu. 2018. Neural User Response Generator: Fake News Detection with Collective User Intelligence.. In *IJCAI*. 3834–3840.
[42] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *EMNLP*.
[43] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. *arXiv preprint arXiv:1507.04808* (2015).
[44] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues.. In *AAAI*.
[45] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *ACL*.
[46] Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. 2016. Hoaxy: A platform for tracking online misinformation. In *WWW*.
[47] Prashant Shiralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2017. Finding streams in knowledge graphs to support fact checking. In *ICDM*.
[48] Kai Shu, Suhang Wang, Thai Le, Dongwon Lee, and Huan Liu. 2018. Deep headline generation for clickbait detection. In *ICDM*.
[49] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.
[50] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for Fact Extraction and VERification. In *EMNLP*.
[51] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *ICML* (2015).
[52] Nguyen Vo and Kyumin Lee. 2018. The rise of guardians: Fact-checking url recommendation to combat fake news. In *SIGIR*.
[53] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *ACL*.
[54] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.
[55] Wenjie Wang, Minlie Huang, Xin-Shun Xu, Fumin Shen, and Liqiang Nie. 2018. Chat More: Deepening and Widening the Chatting Topic via A Deep Model. In *SIGIR*.
[56] William Yang Wang. 2017. " Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *ACL*.
[57] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *KDD*.
[58] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y Zhao. 2017. Automated Crowdturfing Attacks and Defenses in Online Review Systems. In *SIGSAC*.
[59] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW*.