Project Report: Generating Fake Yelp Restaurant Reviews with RNN

Bowen Li
ECE. Tandon School of Engineering
New York University
Brooklyn, US
bl2305@nyu.edu

Sui Huang
ECE. Tandon School of Engineering
New York University
Brooklyn, US
sh4507@nyu.edu

***Abstract*—This document is the report for our final project of EL9163 *Cyber Security in Machine Learning* class. In this project, we dug deep into the concept and realization of a special kind of Recursive neural network - Long Short Term Memory network. In this document, we tried to keep a record of our method to analyze and resolve the problem of generating fake reviews for different kinds of restaurant.**

First of all, we reproduced the attacking method in paper *Automated Crowdturfing Attacks and Denfenses in Online Review Systems*[1]. After analyzing its results, we came up with a new idea of training distinguish models for different restaurant. Then we took Chinese, Japanese and Mexican restaurants as examples to display the effectiveness of our methods. At last, a brief discussion ended our project and left us more to explore.

Since this project in using a huge dataset and neural networks, we don't recommand reproducing it without a powerful GPU, or even, without powerful GPUs. It took us more than two weeks to finish all raw training procedures with a GTX 1070, including re-trainings caused by mis-cleaned data. Both our thoughts, results, and a few pieces of key codes were included. Our presentation slides were also appended.

***Index Terms*—Recursive Neural Network; Web Security; Fake Review; Natural-Language Processing;**

I. Introduction

While people are eager to try new restaurant around, they would like to reduce the risk of stepping into a wrong place and enduring the consequence of a imprudent choice. What come into their mind is finding a great recommanding application. *Yelp*, with its coverall database and real-human reviews, might become their perfect choice. And this is where a crowdturfing attacker find his way to sneak in. Thus the Internet will no longer be a reliable information source but a hidden marget where hackers make their profits.

There is a saying, *Keep your friends close and your enemies closer.*. Trying to think like an attacker would help us learn to defend. So we are trying to figure out a least expensive way passing our fake information off as genuine. Using machine generated fake reviews is the best way to do it. But generating fake paragraphs is not a classification or prediction problem, a tradtional machine learning model can no longer satisfy this special request. So Recursive Neural Network is the model we need. Its can remember the 'relationship' between characters and imitate the trainning text we feed.

Since it had been discussed in class how to generating text from existing text and we also got the paper *Automated Crowdturfing Attacks and Denfenses in Online Review Systems*[1]. What we want to do is make it more customize, or more specific. We are looking for a effective way to generating fake restaurant reviews, targeting different kinds of restaurants. And of course, there would be 1 star bad reviews to 5 star good reviews, we also want to split the fake reviews in more detailed classes. That is, good/bad reviews for different kinds of restaurants.

The first method came into our mind is, of course, trying the methond in that paper. Trainning an general model, creating initial reviews, then replace the words with other words related closely to the specific kind of restaurant. So we first tried a identical model generating from that paper. It was a huge workload so we stopped after generating good reviews. And also the results was not that ideal. Simply Replacing words can sometimes lead to ambiguity.

So we finally came up with our own method.

II. Preliminaries

In this section, we will discuss how to clean the yelp dataset and introduce the basic knowledge about RNN networks.

A. Yelp Dataset

B. RNN

III. Attack Methodology I

This attack methodology is based on the paper *Automated Crowdturfing Attacks and Denfenses in Online Review Systems*[1].

A. Attack Methodology

B. Attack Procedure

a) Data Preprocessing:

b) Generating Initial Reviews:

c) Review Customization:

Codes below were downloaded and edited from this Gitgub: https://github.com/ajmanser/Yelp [5]

The food_related function computed the similarity of our input *nouns* with words in *WordNet*. If somes words from WordNet have high similarity(¿0.2) with out input, then we keep a record of those words.

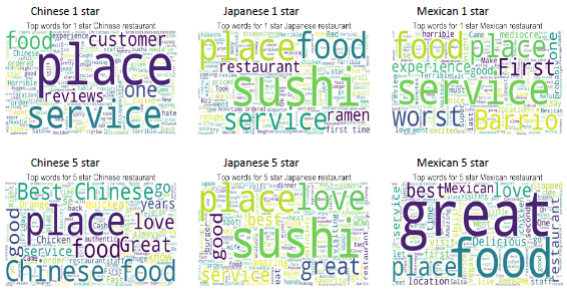

Fig. 1. Frequency based WordCloud Results

The review_to_nouns function was used to clean the *initial review* generated from our general training model.This function extract the nouns from initial review.

The personalized_clean_up function was used to replace words.If there was a noun, extract using review_to_nouns function exixts in the result from food_related function, we replaced this noun with our *input*. This was what we called 'customize'.

```
# nouns is the inpur seed
def food_related(nouns):
    food=wn.synset('food.n.01')
    final_list=[]
    for word in nouns:
        temp=word
        word=word+'.n.01'
        try:
            if food.wup_similarity
               (wn.synset(word))>0.20
                and temp!='food':
                final_list.append(temp)
        except:
            pass
    return final_list

def review_to_nouns(review):
    is_noun = lambda pos: pos[:2] == 'NN'
    token=nltk.word_tokenize(review)
    nouns=[word for (word, pos)
           in nltk.pos_tag(token) if is_noun(pos)]
    return nouns

def personalized_clean_up(review,user_items):
   generic_nouns=review_to_nouns(review)
   food_generic=food_related(generic_nouns)

   user_picked_items=user_items.split(",")

   final=[]
   for word in re.findall(r"[\w']+|[.,!?;]",review):
        if word in food_generic and
        len(user_picked_items)>1:
            word=random.choice(user_picked_items)
            final.append(word)
        else:
            final.append(word)

    new_review=" ".join(final)
    return re.sub(r'\s+([?.!",])', r'\1',
                  new_review)
```

d) Trainning:

C. Results(5 star good review only)

a) Chinese:

Seed: "sushi,salmon,tuna"

Review:The staff were very friendly and attentive. Very fast and friendly service and great tuna.

Review:This sushi has some of the best tasting in west sushi. It's a tuna and fun the next day of going. Very busy but nothing comes close to sushi in the kitchen.

b) Japanese:

Seed: "dumpling,noodle,tofu"

Review: This tofu is amazing! The food was delicious!!! I was there for a game and on our dumpling break for 12. 99 and I love it. For noodle was also excellent

Review:dumpling poutine! Been going to Zo tofu for years. We always have a great time there.

c) Mexican:

Seed: "taco,burrito,guacamole"

Review: taco food and great price! Something different from the convenience of the taco but homemade guacamole food is pretty good.

Review: Loved this taco. Loved it. Service was excellent, price was right and the atmosphere was kinda high! Can't wait to go back! Good taco to dine with friends!

IV. Attack Methodology II

A. Attack Methodology

a) Trainning:

```
model = keras.models.Sequential()
model.add(layers.LSTM(512, input_shape = (maxlen,
          len(chars)),return_sequences = True))
model.add(layers.LSTM(512, input_shape = (maxlen,
          len(chars))))
model.add(layers.Dense(len(chars),
          activation='softmax'))

optimizer = keras.optimizers.Adam(lr=0.001)
model.compile(loss='categorical_crossentropy',
              optimizer=optimizer)
```

B. Attack Procedure

a) Data Preprocessing:

b) Generating Fake Reviews:

C. Results

a) Chinese:

(5 star) This is my new favorite Chinese place in Vegas. The servers are friendly. We had the chicken pakora, chicken wings, and shrimp with broccoli is AMAZING too. We love it!

(1 star) The food was terrible. Worst chinesefood ever!!! The service was really bad, the food was horrible. I wouldn't recommend this place to anyone.

b) Japanese:

(5 star) This is a great sushi place in the world! Sushi is amazing, the price is reasonable and the prices are perfect. The staff is super friendly and the service was very even better. Happy hour selection was great and the food is fantastic. I will definitely be back.

(1 star) Poor attempt at Japanese noodle .The food is decent, but I'd rather give my order. Food was mediocre at best.

c) Mexican:

(5 star) The best Mexican food in Vegas. The food is amazing and the salsa bar is very good.

(1 star) Sat down excited to eat here. Been there into top munright mexicanfood. Over priced for what you get. Will not be returning to this place anymore!

V. Defense

a) A supervised ML scheme based on linguistic features (feature filter):

First, we label all reviews with genuine and fake, and using RNN to learn how to tell a difference. But the result is not good. After dozens of epochs of training, the model still cannot distinguish genuine from fake.

b) A plagiarism detector to check for duplications:

We thought the reviews generated by RNN might have some grammar mistakes or statistical spelling anomaly like always repeating using several words. However, this method does not work well. It turns the model we trained using RNN can generate normal text with limited grammar mistakes.

c) Human User Study:

We have to say it's not a smart way to defense the attack but is the most basic and original way one which usually works. However, this time its very hard to say human beings can truly identify genuine reviews from all reviews.

d) NB Classifier:

In the 5/11 poster session, we discussed with one od our classmates about his project, using NB-classifier, as we did in Lab1, to classify fake reviews. This method indicated that we should extract key words(features) from true/fake reviews. His result was not very good. Our opinion was that since the fake reviews were generated from real ones, the most vital features could still be the same. So this method still would not work.

e) Using two RNN and compare their results:

This defense method comes from the paper *Automated Crowdturfing Attacks and Denfenses in Online Review Systems*[1]: *This leverages the fact that a generative language model builds a fixed memory representation of the entire training corpus, which limits the amount of information that can be learned from a training corpus.*

This means we shoudl train two RNNs, based on real and fake reviews. Test under each RNN, the input would have a high probability of generating next letter in one RNN. For example, if an input always earning high probability when generating fake reviews in FAKE RNN, so this might be a fake review. According to the paper, this is currently the most effective way.

VI. Discussion and Conclusion

VII. Library Version & System Configuration

a) Major Library Version:

- Anaconda 5.1
- Jupyter Notebook 5.2.2
- Python 3.6.1
- TensorFlow 1.4.0
- Keras 2.0.9
- Numpy 1.13.3
- nltk 3.2.4

b) System Configuration:

- OS: Windows10
- CPU: Intel(R) Core(TM) i7-6700
- GPU: NVIDIA GeForce GTX 1070

References

[1] Yao, Yuanshun, et al. "Automated Crowdturfing Attacks and Defenses in Online Review Systems." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.

[2] Luca, Michael, and Georgios Zervas. "Fake it till you make it: Reputation, competition, and Yelp review fraud." Management Science 62.12 (2016): 3412-3427.

[3] Mukherjee, Arjun, et al. "What yelp fake review filter might be doing?." ICWSM. 2013.

[4] Yelp dataset challenge 2017. https://www.yelp.com/dataset_challenge. (2017).

[5] https://github.com/ajmanser/Yelp