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 Defenses in Online Review Systems*[1]. After analyzing its results, we came up with a new idea of training different 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 recommend 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 recommending 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 market 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 traditional 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 training 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 Defenses 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 method in that paper. Training an general model, creating initial reviews, then replace the words with other words related closely to a 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.

We thought of training different model for different kinds of restaurant, as well as training different model for good/bad reviews. This seemed to be a better method because this time we are not simply replacing nouns. We are trying to generating reviews from people's thoughts of different restaurants. The nouns are binded tightly to the different training set. So we believe this would have a better results.

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

We use the Yelp Challenge dataset to train the RNN language model, containing a total of 4.1M reviews by 1M reviewers, collectively targeting 144K businesses [4]. In order to reduce time of processing, we downloaded the .csv version of this dataset.

We pre-processed the review text by removing all extra white space and non-ASCII characters. Also, we remove reviews that were not English and some duplicated reviews. Additionally, we separate the reviews in the corpus by the delimiter tokens "¡SOR¿" (start of review) and "¡EOR¿" (end

of review), so the model also learns when to start and end a review by generating these two tokens.

Original Text:

”This place is ok. The guy had the grill too high and slightly burnt most of our food. Sushi looked pretty good, but I didn’t try it. We’ll be sticking with Ah So as their food is much better tasting””łyelpłsashimi don,ŁłłdonłŁŁłłł””bomb ass food, horrible service though””bomb ass food, horrible service though””The food is good, but not worth spending $100 per person. I’ve had better sushi in town for cheaper price”

Edited Text:

”¡SOR¿This place is ok. The guy had the grill too high and slightly burnt most of our food. Sushi looked pretty good, but I didn’t try it. We’ll be sticking with Ah So as their food is much better tasting.¡EOR¿” ”¡SOR¿bomb ass food, horrible service though¡EOR¿” ”¡SOR¿The food is good, but not worth spending $100 per person. I’ve had better sushi in town for cheaper price.¡EOR¿”

B. RNN

For all experiments, we use a Long Short-Term Memory (LSTM) model [6], an RNN variant that has shown better performance in practice. The neural network we used contains 2 hidden layers, each with 1,024 or 512 hidden units.

For training, the input string is split into batches of size 256. Training loss is computed using cross-entropy , and weights are updated using Adam optimization, a common optimization technique for neural network training.

```
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)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 60, 512)           1722368
_________________________________________________________________
lstm_2 (LSTM)                (None, 512)               2099200
_________________________________________________________________
dense_1 (Dense)              (None, 328)               168264
=================================================================
Total params: 3,989,832
Trainable params: 3,989,832
Non-trainable params: 0
_________________________________________________________________
```

Fig. 1. Model Summary

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

First we used a frequency based word cloud to see whether it is possible to only replace nouns in each fake reviews. The result was shown in the picture below.

Fig. 2. Frequency based WordCloud Results

B. Attack Procedure

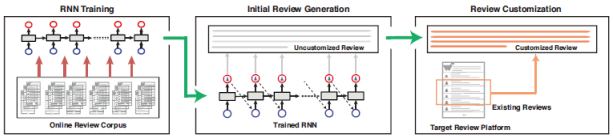

Fig. 3. Model flowgraph

C. Data Preprocessing

Get reviews from yelp dataset with 5 star. And then cleaned the data as we discussed above.

D. Generating Initial Reviews

Using ramdom sentence to generate initial reviews.

For example:

Seed: The food is good and the service

Result:The food is good and the service was even better. The food is good and you can’t beat the price. Friendly service. I would recommend this place to anyone who enjoys.

E. Review Customization

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

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

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 exists in the result from food_related function, we replaced this noun with our *input*. This was what we called ’customize’.

```
# nouns is our input words
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)
```

F. Training

Did this with a GPU.

G. Results(5 star good review only)

a) Chinese:

Input: "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:

Input: "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:

Input: "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

Our own method of generating fake reviews.

A. Attack Methodology

This time, we were training

B. Training

C. Attack Procedure

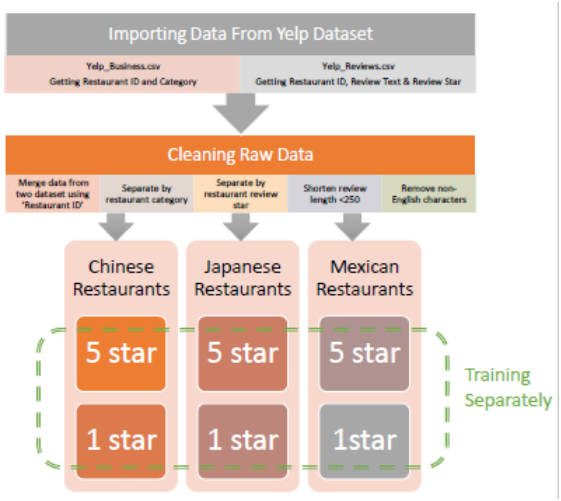

Fig. 4. Our own model flowgraph

D. Data Preprocessing

we split reviews into three groups, according to the restaurant kinds. And them extract the 1 star and 5 star reviews, as the bad/good review traning set.

	business_id	name	categories	stars	text
0	PfOCPjBrlQAnz__NXj9h_w	"Brick House Tavern + Tap"	American (New);Nightlife;Bars;Sandwiches;Ameri...	4	I've always enjoyed my time at brick house foo...
1	PfOCPjBrlQAnz__NXj9h_w	"Brick House Tavern + Tap"	American (New);Nightlife;Bars;Sandwiches;Ameri...	2	1st time here. Came w my Unc bc Louies was clo...
2	PfOCPjBrlQAnz__NXj9h_w	"Brick House Tavern + Tap"	American (New);Nightlife;Bars;Sandwiches;Ameri...	1	Worse service ever andI use to be a server so ...
3	PfOCPjBrlQAnz__NXj9h_w	"Brick House Tavern + Tap"	American (New);Nightlife;Bars;Sandwiches;Ameri...	5	I am updating my review to 5-stars because I t...
4	PfOCPjBrlQAnz__NXj9h_w	"Brick House Tavern + Tap"	American (New);Nightlife;Bars;Sandwiches;Ameri...	4	I enjoyed this place. I went the night the Bu...

Fig. 5. Ectract Reviews from dataframe

E. Generating Fake Reviews

Seed sentence we used was quite simple: "'This is a text file. This file is just something to help the model get started! You can use anything.'" This sentence won't influence the final fake review because we were catching relationships between characters insead of words.

F. 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 of 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 Defenses 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 should 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

A. Train a general model then customize the output

For the first method, its model needs lots of data and time to gain good enough parameters, which might possibly have the same performance compared to the second one. However, once the training session is done, all it needs is a little customization using the functions we listed above. This general model can deal with most problems including some corner cases like some rare and specific kind of restaurant.

As the result above shows, it might come up with some review that are not 'that' readable. But since this model is able to generate 'general case reviews' in a very efficient way, one can always find out proper customize way to replace the nouns. So, this is still a good method.

B. Train a customized model to generate customized reviews

For the second method, it has very good pertinence because it is designed for a specific kind of restaurant. So, its model size can be significant less than the first one and it usually takes less data and time to finish the training process. And the result is better than the first one. We asked some of our family members about what they think of these fake reviews, and they admitted that it was hard to tell which one is genuine or fake.

However, the size of training dataset become a problem here. When the data it needs is insufficient, like Vietnamese restaurants, they might have only hundreds of reviews in Yelp dataset, it may not be easy to find enough genuine reviews for training. Thus, the model might not even converge and a few 'emoji' can ruin this whole training procedure. So, using the second method, the dataset should be big enough and well-managed before training.

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

ACKNOWLEDGMENT

Many thanks to our professor and his amazing lectures. Thanks to the perfect paper *Automated Crowdturfing Attacks and Defenses in Online Review Systems*[1]. Thanks to the gitbub repository https://github.com/ajmanser/Yelp. This project could not be done without any of them.

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
[6] Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 17351780.

.