DS-GA 1001 Term Paper

Narshion Ngao

SCT-C004-3717-2017 Msc. Computer Systems - 2018

Jomo Kenyatta University of Agriculture & Technology

Course: ICS 3210 - Information Systems Security and Audit

March 22, 2021

Abstract

This is a review of journals in the area of using Artificial Intelligence for Intrusion Detection and Prevention Systems. The journals have discussed the milestones that have been achieved in the area of network security particularly intrusion detection. They has also highlighted several Machine Learning algorithms that can be applied in this area to improve the IDS systems. Four articles have been reviewed titled as follows: -

1. Automatic Detection and Correction of Vulnerabilities using Machine Learning by Robin Tommy and others.
2. Machine Learning Classification Model For Network Based Intrusion Detection System by Sanjay Kumar and others
3. Preventing Attacks and Detecting Intruder for Secured Wireless Sensor Networks by Gauri Kalnoor and Jayashree Agarkhed
4. Web Application Security: Threats, Countermeasures, and Pitfalls by Hsiu-Chuan Huang and others.

The following text describes a brief overview of what these articles have reviewed in this field.

1 Business Understanding

By Wikipedia, Yelp serves as a business directory service and crowd-sourced review forum. Every day thousands of reviews are generated by Yelp users, recording their experiences at facilities like restaurants,

retailers, and pharmacies. Yelp has set up a star rating system accompanied with users' reveiews for the customers to share their experience and feedbacks. According to Statista[1], yearly generated yelp reviews has been growing exponentially, and finally hit as many as 177 million in the year of 2018. Therefore, appropriately analyzing review data might generate unexpected business values.

Among all types of analysis, one of the big issues is analysing customer sentiments. With the sentiment data, we could easily track the review of restaurants and predict customer trends. In this way, business could adjust to the present market situation and satisfy customers in a more consistent, accurate and considerate way. Once the business get the current customer trends, they could easily develop more captured menus or strategies dynamically. For instance, decision-makers could know what is being properly implemented and what needs further improvement.

One further application of sentiment data is might use the model to derive features that are correlated to the sentiments of reviewers. By using generated features one can generate business suggestions in improving qualities of services, products, or reducing selling prices, which might give business a higher opportunity to have higher short-time or long-time profits.

It is intuitive accepted that Yelp reviews reflect customer sentiments. However, only part of the intuition is correct. By looking at a few examples of 5/5 rating reviews and 1/5 rating reviews, we find that most of them represent absolute positive sentiments and negative sentiments respectively. However, in more intermediate classes, including 2/5, 3/5, and 4/5 star ratings, it is a more different case. Consider that all reviewers are subjective, it is possible that their grading standard differ. For example, some of them might take 3/5 rating as a bad score, while others treat it positively. A more possible situation is that this is a complex of positive and negative sentiment. Therefore, relying simply on ratings is not adequate to get user sentiments. A more reliable way is to get user sentiments from the text.

As a result, our data mining task is to build a model that analyses text data, in order to get the correct sentiment class, especially for ambiguous data like 3/5-rating reviews. We simplify this task to be a binary classification problem, i.e. between "positive" and "negative" classes, considering the complexity of the data we have mentioned above. This task guides our data selection, labelling, and modelling. Finally, we present the results of models, possible applications, and potential improvements to achieve further goals.

2 Data Overview

• Identify and describe the data (and data sources) that will support data mining to address the business problem. Include those aspects of the data that we routinely talk about in class and/or in the homework.

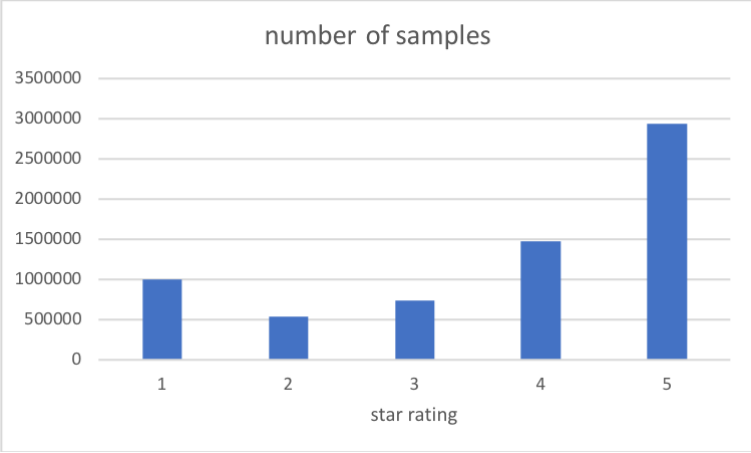
• Specify how these data are integrated to produce the format required for data mining. • Give a clear and precise definition of the target variable. • Make a summary of any feature engineering that should be performed, which may include binning, non-linear transformations and domain knowledge based feature extraction.

2.1 Data Source

The Yelp review dataset is acquired through Yelp Dataset in its data challenge in the following website:https://www.yelp.com/dataset/documentation/main. The dataset contains 6685900 instances, each of which has the following variables: (review_id, user_id,business_id, stars, date, text, useful, funny, cool). It includes customers' reviews and star ratings of different restaurant, ranging from 2014 to 2018. The following parts we discuss the distribution of some important features,identify what variables are selected as input features and how target variable is defined.

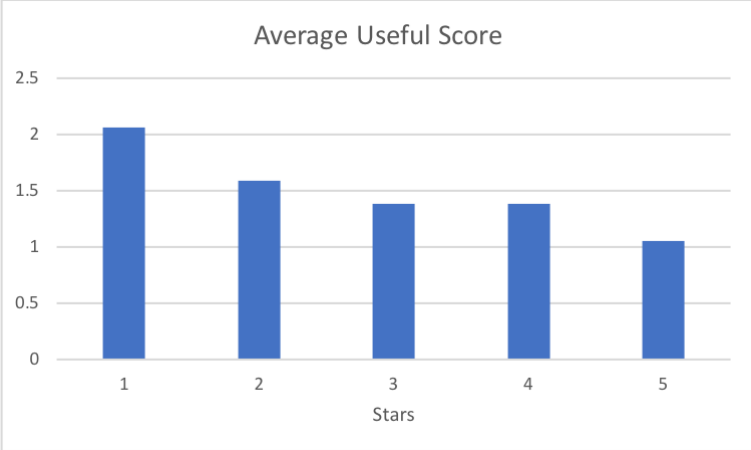
2.1.1 Exploratory Data Analysis

In terms of star rating, most of the customers give 4 or 5 stars, occupying over 65 percent of the whole instances.



However, the reviews of star 1 and 2 receive votes for 'useful' more than 4 and 5, which indicates that

most people actually find the negative reviews reveal more useful information than positive reviews.



2.2 Data Preparation

2.2.1 Target Variable Construction

The target variable that we are trying to predict is the sentimental attitude reflected from the review text data, which is a binary variable:

$$Y = \begin{cases} 1 & \text{if sentiment is positive} \\ 0 & \text{if sentiment is negative} \end{cases}$$

Our dataset has no explicit target variables, but the variable "stars"(ranged from 1 to 5) can help derive the target variable since the star rating for each review would mostly reflect how people feel towards the restaurants, so we decide to choose star rating and transform it to our binary target variable (positive =1, negative =0).

For the credibility of the transformation from star rating to sentiment label, we only select the samples with stars = 5 and stars = 1, because we believe that reviews with stars rating of 5(1) are most likely to be positive(negative). We then denote "stars = 5", "stars=1" as positive sentiment(Y=1),negative sentiment(Y=0),respectively.

2.2.2 Subsampling

Considering the large dataset which is not feasible to be operated on the local machine, we adopt Pyspark where we can implement sql query to preprocess the data.

We select instances with "useful" greater than 2 because we want to filter out reviews that may not give useful information about their experience or comments, while at the same time, we want to keep more data. This reduces our final dataset to 788347 instances.

2.2.3 Text Preprocessing

Before modeling, we preprocess our text data so that it is more digestible form so that our machine algorithm can adjust and perform better. Note that in our deep learning model, we ignore steps in "Stopwords removal" and "Stemming" in our RNN-based model, in which we map each word token to an index in range of vocabulary size instead (Paddings are included).We mainly use the pyspark.ml.feature and nltk packages to do the following:

1.Lowercase the text and remove special characters

In the first step, we lower the whole text and remove special character including numbers, because we consider them as meaningless in analyzing the sentiment of each review.

2.Stopword removal

We use StopWordsRemover in pyspark in this step. We remove common words in the sentence such as "we","are","i", and "the" that are not helpful to our analysis. It also saves us a lot of time in analyzing this large volume of text data.

3.Tokenization:

We use the word_tokenize() function in nltk package tokenize each document of review to split the data into small chunk of words. This is an important step before we convert string features into numerical features. Because the format of the dataset became intractable when we do tokenization as the data in pyspark dataframe, we decide to do it in our pandas dataframe.

4.Stemming

We use SnowballStemmer in nltk package in this step. With stemmization, we can return a word in its stem form from a group of words in different forms. As an example:

$$\left.\begin{array}{l} playing \\ played \\ plays \end{array}\right\} play$$

Uptill here, we have complete the data preprocessing step, as an example,here is an complete review from the dataset:

"This place has gone down hill. Clearly they have cut back on staff and food quality.Many of the reviews were written before the menu changed. I've been going for years and the food quality has gone down hill.The service is slow my salad, which was $15, was as bad as it gets.It's just not worth spending the money on this place when there are so many other options."

After preprocessing, it becomes:

"[place, gone, hill, clear, cut, back, staff, food, qualiti, mani, review, written, menu, chang, ive, go, year, food, qualiti, gone, hill, servic, slow, salad, bad, get, worth, spend, money, place, mani, option]"

2.2.4 Feature Engineering

In this feature engineering step, we transform the raw text data into feature vectors. We split the text up into words based on white-space. New features are created by implementing Count Vectors and TF-IDF Vectors from our dataset. Count Vector represents the frequency count of a particular term in the document. TF-IDF Vector means Term Frequency (TF) and Inverse Document Frequency (IDF). It indicates how important a word is to the document and produces a measure that penalizes words that appear frequently, so it considers overall document weight of a word across all documents. This process allows us to compute the probability of each review belonging to each specific class (positive or negative).

3 Modeling & Evaluation

• Discuss choices for data mining algorithm: what are alternatives, and what are the pros and cons? • Identify an appropriate baseline model and report its performance. • Describe an evaluation framework you

will use to improve upon the baseline. • Perform an analysis of possible algorithms and use the data science experimental framework to choose an optimal candidate. • Demonstrate how you were able to improve upon the baseline and document the process of doing so. • Discuss why and how this model should "solve" the business problem (i.e., improve along some dimension of interest to the firm). • Discuss the type of evaluation metric that should be used to choose the best algorithm. How does this metric relate to the business problem?

Since sentiment classification is a binary classification, multiple classification models have been adopted. Among them, naive bayes and logistic regression models have been applied as baseline models.

3.1 Evaluation Metric

Common evaluation metrics for text classification problem including accuracy and the Area Under the ROC Curve(AUC) are used to evaluate the performance of each model. Accuracy could be used as an effective way to judge the best model, since it is the fraction of predictions whether our model got right or not . ROC curve (receiver operating characteristic curve) draws the performance of a binary classification model at all thresholds. AUC measures the entire area underneath the ROC curve from (0,0) to (1,1). Therefore, AUC could comprehensively summary how our classifier separates the reviews being tested into positive and negative in an efficient way. The higher AUC indicates that the classifier is more capable of distinguishing between these two categories.

3.2 Naive Bayes Bernoulli

Naive Bayes Bernoulli model assume that the conditional probability of each piece of evidence occurring with a given class is independent.The conditional probability is given by:

$$P(e_i \mid c) = \frac{\text{count}(e_i, c)}{\text{count}(c)}.$$

where $\text{count}(e_i, c)$ is the number of documents in a given class that contain feature e_i and $\text{count}(c)$ is the number of documents that belong to class c. Naive Bayes Classifier works by calculating the conditional probabilities of each feature, e_i, occurring with each class C, as

$$P(c \mid E) = \frac{p(e_1 \mid C) \cdot p(e_2 \mid C) \cdots p(e_k \mid C) \cdot p(C)}{p(E)}.$$

We tune the hyper-parameter C and to improve its performance and get the highest AUC of 0.8117 with

alpha = 1(default setting), and the accuracy of 0.7217.

3.3 Logistic Regression

Logistic regression is a useful and common discriminative model for binary classification problem. It is a generalized linear model that predicts the probability of occurrence of an event with sigmoid function as link function. The parameters are learnt through maximum likelihood.

It models the probability of membership in the positive class as a function of X

$$P(y = 1 \mid x) = f(x) = \frac{1}{1 + e^{-W^T x}}$$

The best model receive the highest AUC of 0.8822 at C = 0.001 with L2 regularization method, with a accrucy of 0.8049

3.4 Random Forest Classifier

As a class of parallel ensemble, random forest combines multiple decision trees, each of which is a low bias, high variance base classifier, to produce an aggregated result and reduce overall variance of prediction.

With hyperparameter tuning on the following hyperparameters: max_depth and min_samples_split which control the complexity of each base classifier by grid search, the best model with max_depth = 40, min_samples_split = 4 and max_features = 'sqrt' gives accuracy of 0.8219 and AUC of 0.9282 on test set.

3.5 Bi-LSTM

Under the idea of Neural Network there are a number of state-of-the-art(SOTA) models which outperform traditional machine learning models in classification problems. The idea of Neural Network is similar to logistic regression, with the only difference that multiple layers of perceptrons turn the model into a non-linear classifier, contrary to logistic regression which is a linear classifier.

The basic form of Neural Networks is a Multi-Layer Merceptron(MLP), also named as Fully-Connected Neural Network. However, in this project we do not use this neural network, since it is based on the assumption that all the input features, in this case review tokens, are independent of each other. Theories in semantics and syntax have told us that the meaning of words in a sentence are defined by the context and syntactical functions, i.e. words are related to each other. Therefore, we decide to use a Recurrent Neural

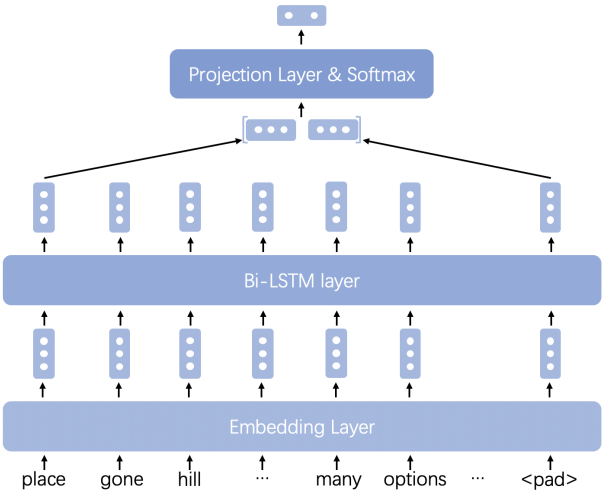
Network(RNN), which is commonly used in finding the representations in sequential data such as audio, text, and some financial data.

In this classification problem we intend to discover crucial information in text that might affect sentiment of reviewer. Therefore, part of information is unnecessary to give the final prediction of sentiment class, such as functional words that only serves in completing syntactic structures. A Long Short-Term Memory Network, one variant of RNN, is capable of doing this job. By using a "gating" approach, LSTM controls the amount of information it accepts, forgets, and outputs given the job it is required to perform. The output of LSTM model, in this sense, is expected to serve as a useful representation of the sentiment of reviewers in our project. Another function of LSTM is that it avoids problems of gradient exploding and vanishing in Neural Networks, which makes the training difficult to converge to its expected performance. We skip further introduction of this function.

Our sentiment analysis model is built as a Bi-LSTM models. Bi-LSTM controls information of the text in both two directions, which gives a more complete representation of the text in sentiment. Our model contains three layers. The first layer is named as word embedding layer, which maps each word into a continuous vector space to derive word-level meaning based on the context. The second layer is a Bi-LSTM layers, which gives us the contextual representation of each position in the sentence. We then subtract the first and last positions, concatenate the two vectors view them as the summary of the text's information related to sentiment. The last layer is a linear function which maps the summary vector to a two-dimension vector and take a softmax function(in two dimension we can also call it a sigmoid function) and obtain the probability of the two classes of sentiments. We reference the paper of Wang et al.[1]. The diagram of our model is presented below. We will give a more detailed introduction of the model layers in the subsequent section.

3.5.1 Layers

Our model contains three layers: word-embedding layer, Bi-LSTM layer, and projection layer. As we have mentioned above, word-embedding layer intend to map tokens from one-hot encodings to a continuous vector space, i.e. word embeddings. Note that a one-hot encoding refers to a vector whose dimension equals to the size of vocabulary and is equal to one in the dimension of the word's index and zeros in other dimensions. Continuous vector representation is based on learning, which learns word vectors based on the context, in order to derive semantic representations based on the context. In constructing word embeddings we have two options: to train word embeddings by ourselves or to use pretrained word embeddings trained or other corpora and provided on the internet. Relative works including glove[2] and fasttext[3]. After experimentation we find



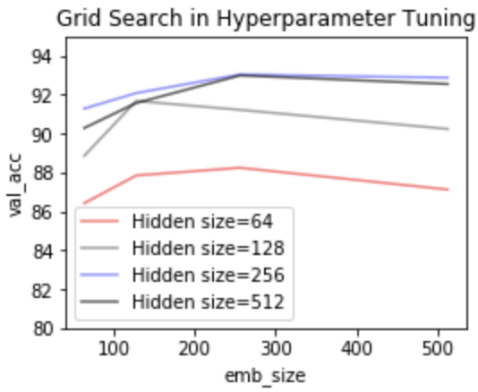
that pretrained word vectors have little improvement in our model performance, but it costs much storage space to store pretrained word embeddings. Finally we decide to train word embeddings by ourselves, and the layer contains only one linear transformation(or affine transformation).

The second layer is an Bi-LSTM layer. Since we have introduced much of the function of LSTM, we will cut this part short and only show our implementation. We use PyTorch to contrust a two-layer Bi-LSTM,i.e. to set "layers=2" and "bidirectional=True" in "nn.lstm", and subtract the first and last position's vectors, as we assume that these two representations are adequate to represent the information for sentiment of an whole review instance. We believe that bidirectional LSTM are better in presenting the review's information than single-direction LSTM since it controls information flows in two ways. We do not do hyper-parameter tuning for the number of layers and the number of directions.

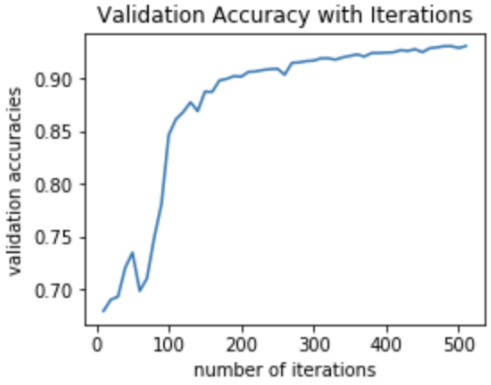
The last layer is only a linear transformation from the concatenation of the first and last position representations to the two-dimensional vector which is followed by a softmax function. The softmax function is used to obtain the probability classes in classification problems. Note that in our case the softmax function is identical to the sigmoid function in logistic regression.

3.5.2 Loss Function and Optimizer

We use Cross-Entropy Loss in our project. For optimizer we use Adam optimizer[4]. In practice, we observe that by using Adam optimizer the model converge much more faster in terms of loss, but the model performance is slightly lower some other optimizers like Stochastic Gradient Descent[5].



(a) 1a



(b) 1b

Figure 1: plots of....

3.5.3 Hyperparameter Tuning

In our project we tune two hyperparameters, combine them, and perform grid search. We tune learning rate and the size of word embedding dimension . We let the dimension size of the output of the Bi-LSTM layer to follow the dimension size of word embeddings, so their number is also tuned. We try learning rate in [0.1, 0.01, 0.001, 0.0001] and embedding size in [64, 128, 256, 512]. The plot below shows the performance of each combination of hyper parameters, and we get the best result of validation accuracy 0.93041 by setting learning rate to be 0.001 and embedding size to be 256. Other hyperparameters are set, in which Bi-LSTM layer dropout rate is 0.2, batch-size is 1,024, and we generate indices for the most 50,000 tokens, turning the rest into unknown tokens. We also present the accuracies as iterations going on, and the model converges after only 3 epochs.

4 Discussion

4.1 Test on Unlabeled Biased Data

We have assumed that Yelp ratings not informative of customer sentiments when it comes to intermediate levels. To test our assumption, we pick several instances, test them on our Bi-LSTM model, and generate the final probabilities of the two classes. Most of the testing results conforms our intuition of the reviews' sentiments. We present 2 instances and their predicted probabilities of being positive reviews. The instances is presented in the following chart.

#	Review	pos_prob
1	Beer, beer, and more beer. It is definitely a low-key nieghborhood bar. As an out of towner, I enjoyed it. It felt like Pittsburgh s version of Cheers except grimier. Unfortunately, the kitchen closes fairly early on weekdays otherwise this place may have received another star.	**0.88277981**
2	Well this was the first time trying this place since they just opened not long ago. We wanted to try something different so we ordered the half sub and soup special. I thought it was expensive if you ask me. It was like $9. So it was over $20 with 2 specials and a drink. It didn't even fill you up for that kind of money! The soup comes in a small container and the sub is very small for being a half. Very disappointed in how much came on the sub. They make them with their back to you and you can not see the items they put on and they don't even ask what you want on it. They just proceed to make it the way they want. I can honestly say you get so much more for your money at Subway for much less and you can make the sub the way you want and you get very full and sometimes even have leftovers! The soup was hot and had a good flavor but such a small container and they fill just above half. We didn't get much at all. Over all not our favorite sub place. Not happy. Not sure if we would go back here.	**0.36329506**

Figure 2: Caption

Note that intuitively the first instance is viewed as a positive review, and the second instance is viewed as a negative review. For the first instance, the Bi-LSTM model neglects negative words including "low-key" and "unfortunately", giving a seemingly justifiable prediction. The same applies to the second instance, which is intuitively treated as an negative review.

4.2 Limitation of the Task

We have designed a binary classification task. However, in reality treating customer sentiments as absolute positive or negative is excessively categorical. A more moderate approach is to transform this binary classification problem into a 3-class classification problem, in which three classes represent positive, neutral, and negative reviews. However, as instances presented above, simply labelling 3/5 ratings to be the class "neutral" will make the class ambiguous and incorrect. Another approach is to label "neutral" class by hand, which is an exhausting job. Before the coming of a more precise dataset, to set this sentiment analysis problem to be a binary classification seems to be the optimal approach.

4.3 Beyond Yelp Reviews

We have built a model dealing with biased review data, which works well in distinguish the sentiments.By learning on the Yelp dataset, we have developed a model which outputs representations of texts to derive sentiment information. One possible application is to implement this model on other unlabeled texts, e.g. reviews on testing the quality of snacks, which is similar to the developing idea "Transfer Learning."

Another possible application is to give business suggestions in improving the quality of food, services, and other aspects. One possible approach is to use TF-IDF values to generate keywords. However, this approach is unrelated to the target sentiments. Another approach is to take in the attention mechanism which is also mentioned in the paper we reference. This improves the interpretablity of the our machine learning model. The attention mechanism will possibly weight important tokens more when generating customer sentiments. By taking tokens of higher attention weights, we might get crucial information in customer reviews, which generates business suggestions based on these crucial tokens. However, since traing an attention-based Neural Network cost more time, and generating useful tokens is not guaranteed, this work is better put in future with careful consideration.

5 Conclusion

• Discuss how the result of the data mining will be deployed. • Discuss how it should be monitored and evaluated in an actual production system. • Discuss any issues the firm should be aware of regarding deployment. • Are there important ethical considerations? • Identify the risks associated with your proposed plan and how you would mitigate them.

References

[1] Statista, *Cumulative number of reviews submitted to Yelp from 2009 to 2018 (in millions)*
https://www.statista.com/statistics/278032/cumulative-number-of-reviews-submitted-to-yelp/

6 Contribution